VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Darius KULAKOVSKIS

RESEARCH OF METABOLIC P SYSTEM FIELD PROGRAMMABLE GATE ARRAY IMPLEMENTATION

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
ELECTRICAL AND ELECTRONIC ENGINEERING (T 001)

VGTU
LEIDYKLA TECHNIKA
Vilnius 2019

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2014–2019.

Supervisor

Prof. Dr Dalius NAVAKAUSKAS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The Dissertation Defense Council of Scientific Field of Electrical and Electronic Engineering of Vilnius Gediminas Technical University:

Chairman

Prof. Dr Vytautas URBANAVIČIUS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

Members:

Prof. Dr Algirdas BAŠKYS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001),

Prof. Dr Habil. Gintautas DZEMYDA (Vilnius University, Informatics Engineering – T 007),

Dr Ofer HADAR (Ben-Gurion University of the Negev, Israel, Electrical and Electronic Engineering – T 001),

Prof. Dr Jurij NOVICKIJ (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The dissertation will be defended at the public meeting of the Dissertation Defense Council of Electrical and Electronic Engineering in the Senate Hall of Vilnius Gediminas Technical University at **1 p. m. on 31 May 2019**.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.
Tel. +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vgtu.lt

A notification on the intend defending of the dissertation was send on 30 April 2019.

A copy of the doctoral dissertation is available for review at VGTU repository http://dspace.vgtu.lt and at the Library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania) and the Wroblewski Library of the Lithuanian Academy of Sciences (Žygimantų st. 1, LT-01102, Vilnius, Lithuania).

VGTU leidyklos TECHNIKA 2019-010-M mokslo literatūros knyga

ISBN 978-609-476-163-8

darius.kulakovskis@vgtu.lt

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Darius KULAKOVSKIS

METABOLINĖS P SISTEMOS ĮGYVENDINIMO LAUKU PROGRAMUOJAMOMIS LOGINĖMIS MATRICOMIS TYRIMAS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
ELEKTROS IR ELEKTRONIKOS INŽINERIJA (T 001)

VGTU
LEIDYKLA TECHNIKA
Vilnius 2019

Disertacija rengta 2014–2019 metais Vilniaus Gedimino technikos universitete.

Vadovas

prof. dr. Dalius NAVAKAUSKAS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001).

Vilniaus Gedimino technikos universiteto Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo taryba:

Pirmininkas

prof. dr. Vytautas URBANAVIČIUS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001).

Nariai:

prof. dr. Algirdas BAŠKYS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001),

prof. habil. dr. Gintautas DZEMYDA (Vilniaus universitetas, informatikos inžinerija – T 007),

dr. Ofer HADAR (Negevo Ben-Guriono universitetas, Izraelis, elektros ir elektronikos inžinerija – T 001),

prof. dr. Jurij NOVICKIJ (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001).

Disertacija bus ginama viešame Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje **2019 m. gegužės 31 d. 13 val.** Vilniaus Gedimino technikos universiteto senato posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuava.
Tel. +370 5 274 4956; fax +370 5 270 0112; el. paštas: doktor@vgtu.lt

Pranešimai apie numatomą ginti disertaciją išsiusti 2019 m. balandžio 30 d.

Disertaciją galima peržiūrėti VGTU talpykloje http://dspace.vgtu.lt ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) bei Lietuvos mokslų akademijos Vrublevskių bibliotekoje (Žygimantų g. 1, LT-01102 Vilnius, Lietuva).

Abstract

The advancement in the fields of electronics and nature inspired computing, including metabolic P (MP) systems, presents new possible solutions to existing problems, however there are still no implementations of MP systems in field programmable gate arrays (FPGA). Therefore, in this work the problem of lack of knowledge about the quality of MP systems implementation in FPGA together with absence of implementation technique for multiple and effective MP systems is solved. The object of the research is specialized MP system implementations in FPGA that operate in real-time. The main aspects of the research object investigated in the thesis are: implementation quality and techniques.

The aim of the thesis is to offer original FPGA based MP system solutions by creating and investigating real-time metabolic process electronic system used for imitation and testing. In order to solve the stated problem and reach the aim of the thesis the following objectives are formulated: using theoretical results of MP systems and other best practices, offer original solutions for MP system transformation to FPGA structural elements and signal processing schemes; reveal quality characteristics of the transformation based on throughput, complexity and power consumption; create real-time metabolic process imitation and testing electronic system and perform its evaluation experiments.

The dissertation consists of an introduction, four chapters and general conclusions. The first chapter reveals the fundamental knowledge on nature inspired computing, MP system definition and application, and FPGA implementation quality estimation. In the second chapter the quality criteria of calculation accuracy, throughput, resource usage, power consumption and interface complexity are selected for the evaluation of MP system FPGA implementation. New combined MP system quality metric ant its visualisation is also proposed. In the third chapter the common FPGA implementation techniques are adapted for MP systems and new unified technique is proposed. The evaluation of the developed MP system implementations in FPGA is presented in the fourth chapter. The experiments consist of a single MP system implementation using three different techniques and a multiple MP system implementation using two new developed unified implementation techniques.

The main results of the thesis were published in 5 scientific publications: three of them were printed in peer-reviewed scientific journals, one of them in Clarivate Analytics Web of Science database, two articles – in conference proceedings. The research results were presented in 6 scientific conferences.

Reziumė

Elektronikos ir gamtinių skaičiavimų, įskaitant ir metabolines P (MP) sistemas, sričių pažanga suteikia naujų galimybių spręsti esamas problemas, tačiau vis dar nėra sukurta MP sistemų įgyvendinimo būdų lauku programuojamose loginėse matricose (LPLM). Todėl disertacijoje sprendžiama žinių apie MP sistemos LPLM įgyvendinimo kokybę ir našių MP daugiasisteminių įgyvendinimo metodų trūkumo problema. Tyrimų objektas yra specializuota realiu laiku veikianti MP sistema įgyvendinta lauku programuojamoje loginėje matricoje. Disertacijoje tiriami šie su tyrimo objektu susiję dalykai: įgyvendinimo kokybė ir įgyvendinimo būdai.

Disertacijos tikslas – pasiūlyti originalius LPLM technologija grįstus MP sistemos sprendimus, sukuriant ir ištiriant realaus laiko metabolinių procesų imitavimo ir testavimo elektroninę sistemą. Siekiant išspręsti iškeltą problemą ir įgyvendinti darbo tikslą, suformuluoti šie darbo uždaviniai: remiantis MP sistemos teoriniais rezultatais ir pasaulinės praktikos įžvalgomis, pasiūlyti originalią metodiką MP sistemos konceptams transformuoti į LPLM struktūrinius elementus ir signalizacijos schemas; atskleisti šios transformacijos kokybines charakteristikas greitaveikos, sudėtingumo ir energijos sąnaudų kriterijų pagrindu; sukurti realaus laiko metabolinių procesų imitavimo ir testavimo elektroninę sistemą ir eksperimentiškai ją ištirti.

Disertaciją sudaro įvadas, keturi skyriai ir bendrosios išvados. Pirmajame skyriuje pateikiamos žinios apie gamtinius skaičiavimus, MP sistemų apibrėžimą ir jų taikymą, LPLM įgyvendinimo kokybės vertinimą. Antrajame skyriuje yra išrenkami skaičiavimo tikslumo, greitaveikos, LPLM išteklių panaudojimo, galios suvartojimo bei sąsajos sudėtingumo kokybės kriterijai, skirti MP sistemų LPLM įgyvendinimų vertinimui. Taip pat pasiūloma nauja bendros MP sistemų kokybės metrika ir jos grafinis atvaizdavimas. Trečiajame skyriuje esami LPLM įgyvendinimo metodai yra pritaikomi MP sistemoms ir yra pasiūlomas naujas apibendrinto įgyvendinimo būdas. Sukurtų MP sistemų LPLM įgyvendinimų tyrimai pristatomi ketvirtajame skyriuje. Tyrimuose nagrinėjami vienos MP sistemos įgyvendinimai trimis skirtingais metodais ir daugelio MP sistemų įgyvendinimai dviem naujais sukurtais apibendrinto įgyvendinimo metodais.

Pagrindiniai disertacijos rezultatai paskelbti 5-iuose moksliniuose straipsniuose, iš kurių 3 atspausdinti recenzuojamuose mokslo žurnaluose. Iš jų 1 straipsnis paskelbtas mokslo žurnale, įtrauktame į *Clarivate Analytics Web of Science* duomenų bazę. Kiti 2 straipsniai – konferencijų medžiagoje. Rezultatai viešinti 6 mokslinėse konferencijose.

Notations

In General

Text	– emphasis;
Text	– computer program, chip familly and number;
a	– scalar (number);
$\mathtt{A}$	– program (in VHDL) variable;
$\blacksquare_{\text{label}}$, $\blacksquare^{\text{label}}$	– main and supplement labels;
$\blacksquare_i$, $\blacksquare^{(i)}$	– main and supplement index;
$\blacksquare^{\top}$, $\blacksquare^{\perp}$	– maximum and minimum values.

Symbols

E_{MAE}	– mean absolute error;
E_{RMSE}	– root mean square error;
$f^{\top}$	– maximum frequency of (circuit) operation;
f_{val}	– value generation frequency;
ϕ_i	– MP system regulators (fluxes);
G, I	– glucose and insulin substances of IVGTT MP system;
k_{θ}	– coefficient of transformation;

M – MP system construct;
N_{t} – total number of elements of type t;
N_{eqLUT} – total number LUT equivalent elements;
N_{WLI}, N_{WLF} – number of bits for integer and fractional parts;
n – time (sample) index;
P – power consumption;
Π – P system construct;
$Q_{\mathrm{MP}} = \{Q_{\mathrm{t}}\}$ – quality of MP system implementation dependent on a set of quality criteria of type t: A – accuracy, T – throughput, R – resource, P – power, I – interface;
Q^* – transformed implementation quality;
$\mathrm{Res}_{\blacksquare}$ – substance residue;
r_i – MP system rules;
σ_{MP} – dispersion of MP system implementation quality;
θ_{i} – boundary parameter of reference frame i;
w – weights of system parameters;
$x(n)$ – instantaneous MP system substance values.

Operators and Functions

$\equiv$ – equivalence;
$\rightarrow$ – substitution;
$\triangleq$ – definition;
$|\blacksquare|$ – absolute value (modulus);
$\|\blacksquare\|$ – norm (length);
$\widehat{\blacksquare}$ – estimate;
$\overline{\blacksquare}$ – mean;
$[\blacksquare,\ ]\blacksquare$ – opening and closing of membrane;
$\max(\blacksquare)$ – maximum.

Abbreviations

ASIC – application-specific integrated circuit;
BRAM – block random access memory;
BZ – Belousov-Zhabotinsky reaction;
DFG – data-flow graph;
DNA – deoxyribonucleic acid;
DSP – digital signal processor;
FPGA – field-programmable gate array;
HDL – hardware description language;
IOB – input-output block;

IVGTT	– intra-venous glucose tolerance test;
JSON	– JavaScript object notation;
LUT	– look-up table;
MAE	– mean absolute error;
MP	– metabolic P (system);
NII	– nature-inspired intelligence;
NPQ	– non photochemical quenching;
PB	– P system with boundary rules;
PBE	– PB systems with environment;
RMSE	– root mean square error;
SF	– scale factor;
TOCP	– total on-chip power;
VHDL	– very high-speed integrated circuit HDL.

Keywords[1]

IEEE Taxonomy	Part(s)
Circuits and systems	
Circuits	
Integrated circuits – Field programmable gate arrays	**1.5 2.3 3 4**
Computers and information processing	
Computer Science	
Programming	
Automatic programming	**3.1**
Performance analysis	**2 4**
Parallel processing – Pipeline processing	**3**
Software	
Software packages – MATLAB	**1.4 2.2 4.2**
Engineering in medicine and biology	
Bioinformatics	**1**
Medical services	
Medical diagnosis – Computer aided diagnosis	**2.1 3.2**
Mathematics	
Accuracy	**2.2 4.2**
Arithmetic – Fixed-point arithmetic	**2.2 3**

[1] Keywords (assigned to parts) are structured and in-line with the latest IEEE taxonomy (see <https://www.ieee.org/documents/taxonomy_v101.pdf>).

Contents

[1]The annexes are supplied in the enclosed compact disc

Introduction

Problem Formulation

The research field of electronics is rapidly advancing, with an emphasis on parallel computing, as evidenced by increasing investment of such companies as Intel (Byrne 2016). Field Programmable Gate Array (FPGA) is one of the technologies capable of parallel computation in hardware. FPGA manufacturing processes have been advancing for a long time together with other electronics and have reached a point where they are highly capable and relatively affordable (Trimberger 2018).

The rapid advancement of electronics means that there is a need to assess the quality of their new use cases and new implementations. There are many quality criteria that are used to evaluate FPGA implementations, although a holistic approach to quality evaluation and optimization (Abdelouahab *et al.* 2016) may be needed to take into account the specifics of an individual implementation.

Together with the advancement in electronics, the research in biology and nature inspired computing is also increasing and influencing the development of intelligent systems (Siddique, Adeli 2015). Some recent examples of natural computing research fields are the convolutional neural network and membrane computing. The European Human Brain Project (HBP 2018) and the SyNAPSE program in the US (DARPA 2013) are examples of projects that use natural computing.

The new research field of infobiotics (Manca 2013) relies on natural computing, as well as other branches of science such as artificial life and computational synthetic biology. The synergy of these branches is promising to achieve such objectives as:

creation of new generation of nature inspired computers; in silico imitate and research nature and its phenomena; perform calculations using natural resources and processes.

P system takes an important place in nature inspired computing (Paun *et al.* 2010). One of its independent branches – metabolic P system (MP) – is based on the infobiotics theory and describes a metabolic process by using a discrete mathematics framework (Marchetti, Manca 2012).

MP systems were used in many different nature inspired applications in various research fields, discussed in detail in Section 1.3. One of the applications of MP systems is the Intra-venous Glucose Tolerance Test (IVGTT) (Manca *et al.* 2011) that models the glucose-insulin interactions. These MP systems were developed in software using well researched tools described in Section 1.4.1.

The problem arises when hardware implementation of MP systems is considered instead of software. The initial theoretical analysis and proof of possibility to implement MP systems in hardware is available (Guiraldelli, Manca 2015a), however the there are no well researched techniques for this task, especially for multiple parallel FPGA implementations scenario. The implementation in FPGA would allow to offer a novel way of solving problems by combining electronics with the MP system theory. To achieve this, new techniques for transformation of multiple MP systems to FPGA structural elements need to be developed.

The main problem that the thesis solves – lack of knowledge about the quality of metabolic P systems implementation in FPGA together with absence of implementation technique for multiple and effective metabolic P systems.

In order to solve the problem, this main hypothesis was raised and proven: all known IVGTT systems can be generalized in a single unified FPGA implementation to effectively model multiple systems working in real-time.

Relevance of the Thesis

From the point of view of the general needs of the society, diabetes treatment is one of the main isses in heathcare, as it is one of the leading causes of death in the world according to the World Health Organization (2018). Globally there were 422 million people living with diabetes in 2014. This number has increased from 108 million in 1980. This shows that the number of people suffering from diabetes is rapidly increasing. It is also worth noting that among people aged eighteen years and above diabetes prevalence has increased from 4.7 % in 1980 to 8.6 % in 2014 (American Diabetes Association 2014). To take the United States as an example of a single country, 29 million of its people (9.3 % of the population) suffer from diabetes. Of those 29 million, only 21 million people have had diabetes diagnosed. That means that at least 8 million people, representing 27 % of the total number of those with diabetes, have not yet been diagnosed.

The diagnosis of diabetes, as well as monitoring and control of the blood glucose level, plays a crucial role for hundreds of millions of people that suffer from this

illness or other glucose related diseases. Among the most common tests used for diagnosis and monitoring of diabetes are the glucose tolerance tests (NIDDK 2016), one of which is the IVGTT. There have also been many efforts to develop an artificial pancreas system that would allow automated monitoring and medication for diabetes patients. Although these systems have been investigated for more than 50 years, a reliable and clinically acceptable glucose regulator is still not perfected (Cobelli *et al.* 2011).

Now could be the right time to further investigate glucose monitoring devices, considering the advances in the field of electronics. Currently the selection of electronic devices is increasing in such rapidly growing markets as the Internet of things (Whitmore *et al.* 2015) or smart healthcare (Research and Markets 2018). In particular, electronics are being introduced in medicine as a replacement of traditional treatment methods (Khosla 2012).

Sensor based electronic medical devices called artificial pancreas are becoming more popular in diabetes treatment. They are shown to be a safe and effective approach to continuous glucose monitoring and insulin medication, although they can still be improved (Bekiari *et al.* 2018).

Lately FPGA has become an economicaly justified tool used in portable electronic devices. They are getting more affordable and their integration level is rising at a rate that has allowed the capacity of FPGA chips to increase by a factor of $10,000$ since their introduction (Trimberger 2018).

The fact that electronic medical devices are becoming more accessible has an effect on social trends such as self-diagnosis that is appealing to many people (Hynes 2013). This trend of self-diagnosis has widely reaching implications as it promises to improve the quality of life of many people wordwide, especially in places where access to professional medicine is limited.

From a practical point of view, the complexity of glucose-insulin interaction models is a problem when considering their embedded hardware implementation, as discussed in Section 2.1.1. MP system does not use differential equations like most other models, thus it could be an efficient modeling way in real-time applications.

The IVGTT MP system was shown to effectively work in software (Manca *et al.* 2011), but hardware implementation would allow for its application in efficient portable devices and improve its scalability for use in multiple patient scenarios. This requires new real-time MP system hardware implementations with sufficient quality characteristics.

Currently, the research of hardware implementation techniques of MP systems is lacking, although there are hardware implementations of similar membrane computing systems (Nguyen *et al.* 2007). The possibilities of MP system implementation in hardware were theoretically examined (Guiraldelli, Manca 2015b), but techniques for multiple system implementation in a single device or the quality characteristics of such implementations are still not investigated enough. Existing general quality estimation criteria for FPGA, discussed in Section 1.5, are not specifically tailored for MP system implementation and need to be reviewed.

The Object of the Research

The object of the research is specialized metabolic P system implementations in the field programmable gate array that operates in real-time. The main aspects of the research object investigated in the thesis are: implementation quality and techniques.

The Aim of the Thesis

The aim of the thesis is to offer an original field programmable gate array based metabolic P system solutions by creating and investigating real-time metabolic process electronic system used for imitation and testing.

The Objectives of the Thesis

In order to solve the stated problem and reach the aim of the thesis the following objectives are formulated:

1. Using theoretical results of metabolic P systems and other best practices, to offer original solutions for metabolic P system transformation to field programmable gate array structural elements and signal processing schemes.
2. To reveal quality characteristics of the transformation based on throughput, complexity and power consumption.
3. To create real-time metabolic process imitation and testing electronic system and perform its evaluation experiments.

Research Methodology

The current states of the relevant scientific fields are assessed using literature and technology analytical review. The infobiotics theory is applied to mathematically define the metabolic process. The concept of quality and its metrics is used to evaluate hardware implementation and automatization processes. Quality metrics of root mean square error (RMSE) and mean absolute error (MAE) are used for accuracy assessment. For other quality metrics FPGA hardware parameter analysis is used. Visualization of multidimensional quality metrics is applied when comparing different implementations. For the development of FPGA implementation techniques the combinatorics theory is used and the techniques of instruction scheduling, parallelization, fixed point arithmetic and data flow graphs are applied. Additionally, the binary search algorithm and formalization into JSON data structure format are used for the automated implementation. Computer simulation is used as a reference point for the evaluation of laboratory experimentation and measurement processes.

Intra-venous glucose tolerance test data sets for the experiments are constructed from the available previous research data. The simulations are carried out with the use of Matlab R2015b software package. The intelligent electronic systems are implemented in Zynq-7000 FPGA. For their development and experimental investigation Xilinx ISE Design Suite 14.7 together with original developed software tools are used.

Scientific Novelty of the Thesis

1. The first metabolic P system hardware implementation technique for unified efficient implementation of multiple similar systems in FPGA, which allows intra-venous glucose tolerance test calculations to be performed in FPGA on a large scale, was developed.
2. Previously unavailable metabolic P system implementation in FPGA quality estimation criteria and new compound value, which estimates complete system quality, enabling the assessment of the viability of different metabolic P system implementations, was defined.
3. New automated implementation technique for MP systems, which takes into account the calculation accuracy affected by fixed point arithmetic, was developed, allowing faster than before implementation of different MP systems in FPGA that is needed for research of mathematical models such as IVGTT.

Practical Value of the Research Findings

Implementation of MP model of IVGTT based on the proposed technique is created. The MP implementation quality criteria allow to select the needed word length parameters and choose the proper FPGA chip.

The implemented IVGTT system allows to simulate glucose-insulin interactions of multiple patients on the same FPGA chip. The data sets can be switched and simulation restarted based on the input parameters to allow rapid research of glucose-insulin interactions based on MP systems.

The Defended Statements

1. The presented metabolic P system implementation techniques can be applied to implement all six known and investigated intra-venous glucose tolerance test metabolic P systems in field programmable gate arrays with root mean square error not higher than 15% using word length of at least 32 bits.
2. The new metabolic P system combined quality metric and its graphic representation clearly separates the quality of different metabolic P system imple-

mentations in field programmable gate arrays and allows to analyze quality characteristics appropriate for these systems.

3. The offered unified combinative intra-venous glucose tolerance test metabolic P system implementation technique ensures 2 to 3 times higher speed compared to the unified pipelined implementation technique.

Approval of the Research Findings

The research results are published in 5 scientific publications:

- one article is printed in a peer-reviewed scientific journal listed in Clarivate Analytics Web of Science database without impact factor (Kulakovskis, Navakauskas 2016);
- two articles are printed in peer-reviewed scientific journal listed in Index Copernicus database (Kulakovskis 2015; Kulakovskis 2019);
- two publications are printed in other scientific works: both in international conference proceedings listed in Clarivate Analytics Web of Science database ISI Proceedings category (Kulakovskis, Navakauskas 2015; Kulakovskis *et al.* 2016).

The main results of the thesis are presented in the following 6 scientific conferences:

- 3rd international workshop on "Advances in Information, Electronic and Electrical Engineering (AIEEE)", 2015, Latvia, Riga;
- 4th international workshop on "Advances in Information, Electronic and Electrical Engineering (AIEEE)", 2016, Lithuania, Vilnius;
- annual national conferences "Science – Future of Lithuania", 2015–2017, Lithuania, Vilnius;
- national conference "KTU PhD Week", 2018, Lithuania, Kaunas.

Structure of the Dissertation

The dissertation contains: introduction, four chapters, general conclusions, summary in Lithuanian, list of references with separately presented list of publications by the author. The list of symbols, abbreviations, keywords and subject index are presented. The dissertation consists of 118 pages, where: 63 displayed equations, 24 figures, 11 tables, 2 algorithms and 7 examples are presented. In total 115 references are cited in the thesis.

1

Metabolic P System Modeling and Implementation Review

The simplest natural living organism is a cell. It has evolved through natural ways for billions of years to do one simple operation – consume substances, transform them to new substances by chemical reactions and remove excess substances from the cell. This process is a basis for most biological organisms living on Earth. If this process can be used to help model metabolic P systems accurately and efficiently, it can help solve challenging biological and medical problems that are relevant to many researchers, such as diabetes treatment (Manca *et al.* 2011).

In Section 1.1 the general field of research of natural and membrane computing is described. The aim of this thesis is to investigate metabolic P systems that are part of this research field and are described in detail in Section 1.2. The importance of MP systems and their practical uses are highlighted in Section 1.3. The ways of implementation of MP systems in software as well as previous research of hardware implementation are presented in Section 1.4. Finally, the aspects of implementation quality in hardware such as techniques and criteria that are needed for further research into the MP system implementation in hardware, specifically FPGA, are discussed in Section 1.5. The key quality parameters influenced by the design of FPGA and ways of their estimation are also presented.

The research results are published in author publication (Kulakovskis 2015). The main results are announced in national "Science – Future of Lithuania" (Vilnius, 2015) scientific conference.

1.1. Natural and Membrane Computing

Natural computing is a broad research field of computer science. Membrane computing is a separate category of natural computing that focuses of cell membranes as an object of computing. This field has proven to have a potential of solving difficult mathematical problems. Many different variations of the original membrane computing were developed, including the Metabolic P system.

1.1.1. Natural Computing

Natural computing is a research field that includes methods, algorithms, models and other things that are directly based on or inspired by processes that happen in nature.

This research field imitates the processes that occur in nature and tries to adapt them to be useful for computing applications. These natural processes have developed in nature over billions of years and can prove to be an important tool for solving complex mathematical and computational problems.

According to Paun (2000), neural networks, genetic algorithms, and deoxyribonucleic acid (DNA) computing are three main areas of natural computing that are well established and especially in the first two cases are proven to be practically useful. Natural computing, sometimes called nature-inspired intelligence (NII), has a big variety of methods and techniques for different applications. According to Vassiliadis, Dounias (2009) there are five major categories of NII related techniques or approaches:

1. ant colony optimization algorithms,
2. particle swarm optimization algorithms,
3. artificial immune systems,
4. DNA computing,
5. membrane computing.

These NII categories were determined by looking at the increasing number of publications by many researchers worldwide. These techniques and approaches are commonly used in such fields like general optimization problems, industrial applications, data processing (Vassiliadis, Dounias 2009).

Membrane computing, including Metabolic P system, is one of the main types of natural computing with increasing number of research. According to Clarivate Analytics Web of Science statistics, the amount of published membrane computing articles started increasing in 2014 and the there are at least 50 to 80 articles published each year. It is discussed in detail in the next section.

1.1.2. Membrane Computing

Membrane computing was first introduced by Paun (2000) and since then it has evolved substantially. The basic described computing principle was the usage of membranes

to transform some kind of objects. The concept of a membrane is inspired by biology – every living cell has a membrane which is used to allow or deny transfer of chemical substances inside the cell. The same concept is applied in membrane computing. Membranes separate different regions that have some objects placed inside. Then these objects can be transformed, destroyed or transfered to a different region. From this a computing device, called P system, is obtained.

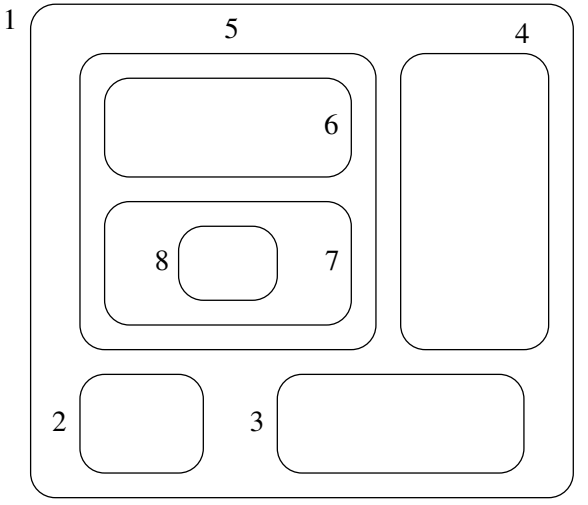


Fig. 1.1. Membrane structure represented by an Euler-Venn diagram, where 1 is a skin membrane and 2–4, 6, 8 are elementary membranes

Fig. 1.1 shows an example of a membrane structure as defined by Paun (2000); Paun, Rozenberg (2002). In it there are different kinds of membranes arranged in a hierarchical structure. Skin membrane is the main membrane which separates the system from the environment. Each membrane produces a region which can contain a number of objects that evolve according to defined rules. An elementary membrane (2, 3, 4, 6 and 8) ends the structure – it has no other membranes inside. In Fig. 1.1 membrane structure was defined by an Euler-Venn diagram, but for easier use it can also be represented by a parentheses expression:

$$[1\ \ [2\ \ ]2\ \ [3\ \ ]3\ \ [4\ \ ]4\ \ [5\ \ [6\ \ ]6\ \ [7\ \ [8\ \ ]8\ \ ]7\ \ ]5\ \ ]1\,. \tag{1.1}$$

A more formal membrane computing device can be defined by a basic type of P system (Paun, Rozenberg 2002). Such P system also has priority relations, ability to dissolve membranes and is represented by this construct as first defined by Paun (2000):

$$\Pi \triangleq \left(V, T, C, \mu, w_1, \ldots, w_m, (R_1, \rho_1), \ldots, (R_m, \rho_m)\right), \tag{1.2}$$

where V is an alphabet of objects; $T \subseteq V$ – the output alphabet; $C \subseteq V - T$ –

catalysts; μ is a membrane structure consisting of m membranes, with the membranes injectively labeled by the elements of a given set H of m labels; m is called the degree of Π; w_i, $1 \leq i \leq m$, are strings which represent multisets over V associated with the regions $1, 2, \ldots, m$ of μ; an evolution rule is a pair (u, v), which are usually written in the form $u \rightarrow v$, where u is a string over V and $v = v'$ or $v = v'\delta$, where v' is a string over $\{a_{\text{here}}, a_{\text{out}}, a_{\text{in}_j} | a \in V, 1 \leq j \leq m\}$, and δ is a special symbol not in V. The length of u is called the radius of the rule $u \rightarrow v$. R_i, $1 \leq i \leq m$, are finite sets of evolution rules over V – each R_i is associated with the region i of μ; ρ_i is a partial order relation over R_i, called a priority relation (on the rules of R_i).

The defined P system is one of numerous classes of P systems. These systems are computationally universal and are powerful enough to solve complex problems. It has been demonstrated that P systems can be used to solve nondeterministic polynomial complete problems in polynomial time (Krishna, Rama 1999; Paun 1999, 2001; Zandron *et al.* 2001). P systems are equal in power and can be used to simulate Turing machines (Alhazov *et al.* 2014; Paun, Rozenberg 2002).

Metabolic P systems can also be defined as a narrow subset of membrane computing. Formally MP systems are considered an extension of P systems, of which there is a number of different types. Let's explore these types of P systems in detail.

1.1.3. P System Types

There are many different types of P systems. The definition of a P systems highly depends on the problem the system is trying to solve. Thus, it can be said, that each application of P system defines a new slightly different type of P systems. Nevertheless, some attempts have been made to classify P systems and attribute them to a particular type. Paun (2010) and Jiang *et al.* (2012) define three main types of P systems: cell-like P systems, tissue-like P systems and neural-like P systems.

Cell-like P systems try to imitate a biological cell in their operation. They rely on a hierarchical arrangement of membranes. Inside these membranes reside other objects which belong to one of the regions. This is called a membrane structure. Then there are certain rules which can be applied on the objects or on membranes. These rules can be of different types, for example multiset rewriting rules or transport rules can be used. Originally, rewriting rules were used by Paun (2000). Later, many types of transport rules were applied, most popularly the symport or antiport rules (Paun, Paun 2002). In cell-like P systems, the membrane structure can evolve and change multiple times, because separate membranes can be destroyed and new ones created. The same can be said about the objects inside the membranes. These objects can also be of different type and therefore have different type of rules applied to them. The basic type of objects can be described by strings, but there are also types of objects which are described by name-value pairs, also called conformons (Frisco 2009), or arrays (Ceterchi *et al.* 2003).

Tissue-like P systems differ from other systems by using several cells with one membrane and putting them in a common environment (Martin-Vide *et al.* 2003).

Evolving objects then can be placed in these cells or in the environment. The cells can communicate through the environment in which they are placed or a direct communication channel can be implemented (Freund *et al.* 2005). In some systems, for example population P systems (Bernardini, Gheorghe 2004), these communication channels between cells can become dynamic and be modifies by specific rules.

Neural-like P systems can be very similar to tissue-like P systems (Martin-Vide *et al.* 2002). Instead of cells, neurons are placed in a graph and contain objects. Also, they have a state which controls their evolution. However, the most popular neural-like P systems are the spiking neural P systems (Ionescu *et al.* 2006). In these systems information is encoded by a sequence of moments or intervals between neuron spikes. Therefore, objects in spiking neural P systems represent a spike which is generated by neuron in specified conditions. Multiple spikes (or P system objects) can be placed in a one membrane cell.

Another approach at classifying P systems was attempted by Krishna (2011), which is based on articles found in a book by Paun *et al.* (2010). The following 7 variants of the basic P system model are defined:

1. The first P system with string objects and rewriting rules as introduced by Paun (2000). A subclass of this P system where replication is used is also mentioned (Krishna, Rama 2001). This variants' advantage is a capability of producing an exponential workspace in polynomial time. This can be used for solving intractable problems.
2. A type of P systems based on biological splicing operation (Head 1987) and called splicing P systems. These systems can be transformed into language equivalent H systems (Paun 1997). Restricted variants of splicing P systems and a universal splicing P system (Frisco *et al.* 2002) are also mentioned.
3. P systems with communication rules (Paun, Paun 2002) don't have any rewriting rules like the first two variants. This type of P system uses communication rules to transfer objects through membranes. These rules are based on biological operations of symport and antiport.
4. Tissue and population P systems are presented as a generalization of cell-like P systems. They are described similarly to tissue-like systems by Paun (2010).
5. P systems with active membranes were the first ones to be able to solve intractable problems in polynomial time. It is mentioned that P systems with active membranes model biological mitosis in cells. Two subclasses of this variant are also presented: systems with dynamic creation of membranes (Arroyo *et al.* 2002) and with mobile membranes (Krishna 2005).
6. Spiking neural P systems (Ionescu *et al.* 2006) here are also considered a separate variant of P systems. They operate with neural impulses called spikes as we have discussed before.
7. The last mentioned variant are P systems with objects on membranes (Brijder *et al.* 2007, 2008; Krishna 2009). In these systems objects can be placed on

the sides of the membranes or have parts in both sides of the membrane (Paun, Popa 2006).

A few extensions of P systems are also presented by Krishna (2011). Dynamic probabilistic P system is one of them (Pescini *et al.* 2006). This system has been used in various biological modeling applications, including genetics and artificial life systems (Paun *et al.* 2006; Romero-Campero, Perez-Jimenez 2008a,b). Another considered extension is Metabolic P system (MP), proposed by Manca *et al.* (2005). This system will be discussed in detail in the next section because it is the object of research of this thesis.

1.2. Metabolic P Systems

Metabolic P systems were developed as a special kind of P systems where the main focus was on the metabolic process that can be observed in all living cells. The metabolic process determines what substances reach the cell through the membrane and what substances are discarded from the cell. Here the membrane acts as a kind of barrier that is used to model various processes and calculations. Let's introduce dynamical P systems as a foundation of MP systems and only then present MP system details and applications.

1.2.1. Dynamical P Systems

During research of different kinds of P systems it was noticed that some adjustments are needed to better model real biological proccesses. It was realized that in biological systems different regions delimited by membranes can interact and depend on each other. Therefore, they must communicate and share information.

Firstly, P systems with symport/antiport rules were developed by Paun, Paun (2002). These systems were discussed in the previous section. Later, a generalization of a mechanism used by this type of P system was considered in Bernardini, Manca (2003b). A new type of rules was proposed, called boundary rules. These rules don't belong to a specific region delimited by membranes, but can interact with multiple membranes and see what happens around their border, hence the name of boundary rules.

P systems with boundary rules (PB) enable more complex membrane interactions and enable the introduction of a new approach to the study of P systems. This new approach is detailed by Bernardini, Manca (2003a), where typical dynamical aspects of biological systems are described by the following four points:

1. Periodicity and quasi-periodicity. Life is based on many biological parameters which are changing periodically and depend on a biological and chemical clock. There are many rhythms and proccesses in life which change with a certain period.

2. Stability and adaptability. Biological systems are resistant to external changes and try to stabilise their state.
3. Growth and degeneration. Three main aspects of life are growing, reproducing and dying.
4. Reproduction and evolution. Forms of life can change and evolve and their dynamics are not limited to a static instance of life.

Periodicity, or oscillation, is named as the most important object of study by Bernardini, Manca (2003a). A real example of such modeling is also provided in the form of chemical oscillators, namely the Belousov-Zhabotinsky (BZ) reaction. This reaction can be described as a metal ion-catalyzed oxidation and bromination of an organic substrate (Zhabotinsky 1991). This reaction is modeled by PB systems with environment (PBE).

A PB system are formally represented by a following construct (Bernardini, Manca 2003b):

$$\Pi_{\text{PB}} \triangleq (V, \mu_0, R, i_{\text{O}}) , \tag{1.3}$$

where V is an alphabet of symbols called objects; μ_0 is the initial configuration where m is the number of membranes; R is a finite set of rules of communication or transformation type; $i_{\text{O}} \in \{1, \ldots, m\}$ is the label of the output membrane.

It has been proven that, when using communication rules of restricted form, PB systems are equal in power to P systems that do not use priority among rules and which do not use any operator for modifying membrane structure (Bernardini, Manca 2003b). It is stated, that power of P systems is increased by bidirectional communication.

PBE systems are formally represented by a following construct (Bernardini, Manca 2003a):

$$\Pi_{\text{PBE}} \triangleq (V, \mu_0, R, E, R_{\text{E}}, O) , \tag{1.4}$$

where V, μ_0 and R are the same variables as in (1.3); E is an environment cycle of period k, environment rules R_{E} are a finite set of rewriting rules on multisets of the form $x \to y$, for $x, y \in V^*$ and O is the label of the observable membrane.

A good example of PBE system is provided by Bernardini, Manca (2003a). The PBE system is adapted to model the Brusselator reaction. The Brusselator is a model of chemical reactions based on the BZ reaction mentioned before. The formulation of this reaction based on the set of following rules is given by Suzuki, Tanaka (1997):

$$\begin{aligned} &\rho_1 \colon A \to X \,; \\ &\rho_2 \colon BX \to YD \,; \\ &\rho_3 \colon XXY \to XXX \,; \\ &\rho_4 \colon X \to C \,. \end{aligned} \tag{1.5}$$

The Brusselator is modeled by PBE system by considering r_1, r_2, r_3, r_4 as resources that the rules ρ_1, ρ_2, ρ_3, ρ_4 use respectively for their application. Objects r_1, r_2, r_3, r_4 are assumed to be produced by the environment with a certain frequency.

In the PBE system formulation by Bernardini, Manca (2003a) the environment cycle E produces a sequence of multisets. By observing the concentration of X,Y in the membrane it is proven by Bernardini, Manca (2003a) that the sequence generated by the PBE system is periodic. Stable oscillations observed by Suzuki, Tanaka (1997) are also confirmed. Formally this PBE system is defined as a following expression:

$$\Pi_{\text{BZ}} = (V^{(BZ)}, \mu_0^{(BZ)}, R^{(BZ)}, E^{(BZ)}, R_{\text{E}}^{(BZ)}, O^{(BZ)}), \tag{1.6a}$$

where

$$\begin{aligned}
V^{(BZ)} &= \{A, B, C, D, X, Y, r_1, r_2, r_3, r_4\}\ ; \qquad (1.6b)\\
\mu_0^{(BZ)} &= [_1]_1\ ;\\
R^{(BZ)} &= \Big\{[_1 r_1 A \to [_1 X;\ [_1 r_2 BX \to [_1 YD;\ [_1 r_3 XXY \to [_1 XXX;\\
&\qquad [_1 r_4 X \to [_1 C;\ A[_1 \to [_1 A;\ B[_1 \to [_1 B\Big\} \cup \Big\{r_i[_1 \to [_1 r_i\ \forall\ 1 \le i \le 4\Big\}\ ;\\
E^{(BZ)} &= \{\beta_1, \ldots, \beta_k\}\ \forall\ k > 0, \beta_i = ABr_j, 1 \le i \le k, 1 \le j \le 4\,;\\
R_E^{(BZ)} &= \emptyset\,;\\
O^{(BZ)} &= 1\,.
\end{aligned}$$

Dynamical P systems show that metabolic P systems were developed from membrane computing and can be considered an extension of P systems. Let's move to the MP system definition.

1.2.2. Metabolic P System Definition

The first theoretical proposal of MP system was made by Manca *et al.* (2005). In this paper it was proposed to observe rewriting rules in membranes from a different viewpoint than in P systems. It is done by defining a metabolic algorithm for computing the evolution of P systems when some initial state and some reaction parameters are given. These parameters can be, for example, reactivities and growing factors. As it is said in the paper, the proposed metabolic algorithm "is inspired by a chemical reading of the rewriting rules":

$$\Delta\|X\| = \sum_{r \in R} \Delta_{\text{r}} \|X\|. \tag{1.7}$$

This equation, as defined by Manca *et al.* (2005), describes the transition from rewriting rules to metabolic equations. To compute the overall molar variation of an object X, contributions of all rules are taken into account by summing up their effects on the concentration of X, where R is a set of rules in P system. An example of rule translation is also provided in the same paper.

The following set of rules:

$$\begin{aligned} &r1\colon AC \to AB\,; \\ &r2\colon BC \to A\,; \\ &r3\colon BBB \to BC\,; \end{aligned} \tag{1.8}$$

associated to coefficients ϕ_1, ϕ_2 and ϕ_3, can be transformed by applying (1.7) to metabolic equations:

$$\begin{aligned} \Delta\|A\| &= +0\cdot\phi_1\|AC\| + 1\cdot\phi_2\|BC\| + 0\cdot\phi_3\|BBB\|\,; \\ \Delta\|B\| &= +1\cdot\phi_1\|AC\| - 1\cdot\phi_2\|BC\| - 2\cdot\phi_3\|BBB\|\,; \\ \Delta\|C\| &= -1\cdot\phi_1\|AC\| - 1\cdot\phi_2\|BC\| + 1\cdot\phi_3\|BBB\|\,. \end{aligned} \tag{1.9}$$

MP system with flux regulation maps is described by Manca (2009). It can be specified by a construct:

$$M \triangleq (X, R, V, Q, \Phi, \nu, \mu, \tau, q_0, \delta)\,, \tag{1.10}$$

where X is the set of substances, R is the set of reactions, V is the set of parameters, Q is the set of states, Φ is the set of flux maps, ν is a natural number which specifies the number of molecules, μ is a function which assigns the mass, τ is the temporal interval, q_0 is the initial state and δ is the dynamics of the system.

MP system without the set Q of states and the dynamics is an MP graph, which can be represented graphically (Manca 2009). When elements τ, ν and μ are also omitted, the resulting thing is called an MP grammar.

1.2.3. Metabolic P Systems for Periodic Function Approximation

Real periodical functions can be approximated using a procedure presented by Manca, Marchetti (2010b). Following are the key elements of the procedure.

To create a new MP system, a real n-dimensional function $f : \mathbb{R} \to \mathbb{R}^n$ can be used. $f[x_0], f[x_1], f[x_2], \ldots, f[x_t]$ is the time series of f along a uniformly distributed sequence of values $x_0, x_1, x_2, \ldots, x_t$. The approximation problem according to Manca, Marchetti (2010b) is to find an MP system such that, for some real vector K, and for some real vectors $\epsilon[i]$ with $0 \le \epsilon[i] \le \epsilon$ and $0 \le i \le t$, the following equation holds:

$$f[x_i] + K = X[i] \pm \epsilon[i]\,. \tag{1.11}$$

Here the vector K provides shift constants for avoiding negative values, vector ϵ is the error tolerated by the approximation, and X is the vector of substance quantities of the system.

The result of the approximation of this function is called MP grammar, discussed in the previous section. The formulation of the problem is very general and has no requirements of a particular MP system structure.

Manca, Marchetti (2010b) claims that this means that the more the MP system is "parsimonious", the better the solution is. To determine the parsimony of an MP system, two features are considered: the reaction stoichiometry and the form of MP system regulators. When a class of regulation functions is chosen, a minimal set of reactions providing the required behavior must be searched to determine the correct one.

Manca (2013) has shown that there is a simple way, directly related to the definition of MP system, to approach the problem by using a mathematical regression method based on the solution of a linear least-squares problem (Luenberger 1979). It is the standard approach to approximate solutions of overdetermined systems, that is, sets of equations in which there are more equations than unknowns. Least-squares means that the overall solution minimizes the sum of the squares of the errors made in solving the equation. A real example of metabolic function approximation is shown in the following section, where the least-squares method is applied for one of the MP systems.

1.3. Applications of Metabolic P Systems

Metabolic P systems most of the time are used for real periodical function approximation. Each approximated function requires a new MP system. There are many fields where the MP system approach to function approximation can be useful, including mathematics, biology, chemistry and other fields. Let's discuss them more thoroughly.

1.3.1. Application in Mathematics

MP system model, called Goniometricus, was defined by Manca, Marchetti (2010b). It is used to approximate sine and cosine functions. The model has two substances S and C approximating in time the evolution of the functions sine and cosine, respectively. Following is the example of the definition of the Goniometricus MP system as presented by Manca, Marchetti (2010b).

First, two equal length time series T_{C} and T_{S} are used. They are relative to the sine and cosine target functions which cover two oscillations (from 0 to 4π) with a step of 10^{-3}. To avoid using negative numbers, each value of the time series is increased by three units:

$$\begin{aligned} T_{\mathrm{C}} &= (\cos(0)+3,\ \cos(0.001)+3,\ldots,\ \cos(4\pi)+3)\,, \\ T_{\mathrm{S}} &= (\sin(0)+3,\ \sin(0.001)+3,\ldots,\ \sin(4\pi)+3)\,. \end{aligned} \tag{1.12}$$

The following relatively simple MP grammar can be used to desribe these functions:

$$\begin{aligned} &r_1 : \emptyset \to C,\ \phi_1 = K_1\,, \\ &r_2 : C \to S,\ \phi_2 = K_2\,. \end{aligned} \tag{1.13}$$

In order to calculate the values of K_1 and K_2, the following system of equations must be solved:

$$\begin{cases} K_1 - K_2 = \Delta C(1)\,, \\ K_2 = \Delta S(1)\,, \\ \dots \\ K_1 - K_2 = \Delta C(n)\,, \\ K_2 = \Delta S(n)\,. \end{cases} \tag{1.14}$$

This system can be solved using these values:

$$\begin{aligned} \Delta C[i] &= T_{\mathrm{C}}[i+1] - T_{\mathrm{C}}[i] \ \ \forall\, i = 1, 2, \dots, |T_{\mathrm{C}}| - 1\,, \\ \Delta S[i] &= T_{\mathrm{S}}[i+1] - T_{\mathrm{S}}[i] \ \ \forall\, i = 1, 2, \dots, |T_{\mathrm{S}}| - 1\,. \end{aligned} \tag{1.15}$$

The system (1.14) has only two unknowns and n equations, with $n = |T_{\mathrm{C}}| - 1 > 2$. The approximate solution can be computed using the least-squares method. The solution of the system, represented by values K_1 and K_2 can be used to calculate the two time series of the residues $\mathrm{Res}_{\mathrm{C}}$ and $\mathrm{Res}_{\mathrm{S}}$ that show the the difference between the approximate solution and the original function at each step:

$$\begin{aligned} \mathrm{Res}_{\mathrm{C}}(i) &= (K_1 - K2) - \Delta C[i] \ \ \forall\, i = 1, 2, \dots, |T_{\mathrm{C}}| - 1\,, \\ \mathrm{Res}_{\mathrm{S}}(i) &= K_2 - \Delta S[i] \ \ \forall\, i = 1, 2, \dots, |T_{\mathrm{S}}| - 1\,. \end{aligned} \tag{1.16}$$

Here it can be observed that the behavior of $\mathrm{Res}_{\mathrm{C}}$ is highly correlated with the evolution of T_{S} and $\mathrm{Res}_{\mathrm{S}}$ is highly correlated with the evolution of $-T_{\mathrm{C}}$. The residue $\mathrm{Res}_{\mathrm{C}}$ goes parallel to T_{S}, while $\mathrm{Res}_{\mathrm{S}}$ goes parallel to minus T_{C}. This means that MP grammar (1.13) needs some corrections to balance the crossing gap. This is done by using a new MP system rule $r_3 : C - S$. It increases S and decreases C at the same time, using a regulator $\phi_3 = K_3 S + K_4 C$ to correctly represent the needed amount. Finally, the complete MP grammar of the Goniometricus is shown:

$$\begin{aligned} &r_1 : \emptyset \to C,\ \phi_1 = K_1\,, \\ &r_2 : C \to S,\ \phi_2 = K_2\,, \\ &r_3 : C \to S,\ \phi_3 = K_3 S + K_4 C\,. \end{aligned} \tag{1.17}$$

1.3.2. Application in Chemistry

From the introduction of MP system, there have been many applications which demonstrate the usefulness of this system. One of the first chemical processes modeled by MP systems was the Belousov-Zhabotinsky reaction (in the Brusselator formulation) (Bianco *et al.* 2006b). As described by Manca (2013), it has the following form:

$$\begin{aligned} a &\to x\,; \\ 2x + y &\to 3x\,; \\ b + x &\to y + d\,; \\ x &\to e\,. \end{aligned} \tag{1.18}$$

MP grammar of Brusselator reaction can be written by simplifying (1.18) reactions by focusing on substances x, y and by considering a, b, d, e as input/output substances, shown in Table 1.1.

Table 1.1. Metabolic P grammar of Brusselator reaction (Manca 2013)

Reactions	Regulators
$r_1 : \emptyset \to x;$	$\phi_1 = 1;$
$r_2 : 2x + y \to 3x;$	$\phi_2 = 10^{-6}x^2y$
$r_3 : x \to y;$	$\phi_3 = 0.03x$
$r_4 : x \to \emptyset.$	$\phi_4 = 0.01x.$

Another significant application of Metabolic P system in chemistry is the Non Photochemical Quenching (NPQ) process. An example of rules and regulators of one particular NPQ MP system are as follows (Castellini *et al.* 2011b):

$$\begin{aligned} r_1 &: c \to o + 12h + p, & \phi_1 &= \alpha_1 - \beta_1 c + \gamma_1 h - \eta_1 x + \nu_1 r + \rho_1 l\,; \\ r_2 &: c \to c + q^+, & \phi_2 &= -\alpha_2 - \beta_2 c - \gamma_2 h + \eta_2 x + \nu_2 l\,; \\ r_3 &: c \to c + f^+, & \phi_3 &= \alpha_3 - \beta_3 x + \gamma_3 l r^{-1}\,; \\ r_4 &: o \to c, & \phi_4 &= -\alpha_4 + \beta_4 o + \gamma_4 h - \eta_4 x + \nu_4 r + \rho_4 l\,; \\ r_5 &: h \to \emptyset, & \phi_5 &= \alpha_5 - \beta_5 c + \gamma_5 h - \eta_5 x + \nu_5 r + \rho_5 l\,; \\ r_6 &: p \to \emptyset, & \phi_6 &= \alpha_6 - \beta_6 c + \gamma_6 h - \eta_6 x + \nu_6 r + \rho_6 l\,; \\ r_7 &: x + 100v \to x + 100z, & \phi_7 &= -\alpha_7 + \beta_7 v\,; \\ r_8 &: y + h \to x, & \phi_8 &= \alpha_8 + \beta_8 y\,. \end{aligned} \tag{1.19}$$

NPQ is a photosynthetic phenomenon that determines how plants accommodate to various environmental light (Manca *et al.* 2009). The NPQ process dissipates excess light, which can be absorbed by the plant in some environmental situations, using non chemical ways (emitting heat). This process is very important for the survival of many

plant species. NPQ MP system has been iteratively improved many times and has numerous variations of MP grammar that have been tested in different cases (Castellini *et al.* 2011a).

In contrast to the BZ reaction, the NPQ process is harder to model. The coefficients used in regulators of (1.19) are:

$$\begin{aligned}
&\phi_1 : \alpha_1 = 7.2389 \cdot 10^{-3},\ \beta_1 = 0.238,\ \gamma_1 = 0.47722,\ \eta_1 = 2.3954,\\
&\quad \upsilon_1 = 6.3515 \cdot 10^{-5},\ \rho_1 = 1.1321 \cdot 10^{-4};\\
&\phi_2 : \alpha_2 = 0.59811,\ \beta_2 = 2.5545,\ \gamma_2 = 32.921,\ \eta_2 = 279.55,\\
&\quad \upsilon_2 = 1.7732 \cdot 10^{-6};\\
&\phi_3 : \alpha_3 = 1.8208 \cdot 10^{3},\ \beta_3 = 8.379 \cdot 10^{4},\ \gamma_3 = 0.90737;\\
&\phi_4 : \alpha_4 = 0.1811,\ \beta_4 = 0.24184,\ \gamma_4 = 0.47722,\ \eta_4 = 2.3954,\\
&\quad \upsilon_4 = 6.3515 \cdot 10^{-5},\ \rho_4 = 1.1321 \cdot 10^{-4};\\
&\phi_5 : \alpha_5 = 8.6799 \cdot 10^{-2},\ \beta_5 = 2.8558,\ \gamma_5 = 5.7365,\ \eta_5 = 28.755,\\
&\quad \upsilon_5 = 7.6423 \cdot 10^{-4},\ \rho_5 = 1.3585 \cdot 10^{-3};\\
&\phi_6 : \alpha_6 = 7.2389 \cdot 10^{-3},\ \beta_6 = 0.238,\ \gamma_6 = 0.47722,\ \eta_6 = 2.3954,\\
&\quad \upsilon_6 = 6.3515 \cdot 10^{-5},\ \rho_6 = 1.1321 \cdot 10^{-4};\\
&\phi_7 : \alpha_7 = 1.0734 \cdot 10^{-4},\ \beta_7 = 3.3287 \cdot 10^{-5};\\
&\phi_8 : \alpha_8 = 5.2981 \cdot 10^{-7},\ \beta_8 = 5.9746 \cdot 10^{-3}.
\end{aligned} \tag{1.20}$$

Ten different substances participate in reactions of the NPQ system. Each reaction also has a regulator expression that uses a list of constants derived from experimental results. In total there are more than 30 different constants in this MP system. This shows that MP systems are capable of simulating processes of varying complexity.

1.3.3. Application in Biology

There are many biological processes that have had their dynamics modeled by MP systems, including the Lotka-Volterra dynamics (Castellini *et al.* 2015) and the Susceptible-Infected-Recovered epidemic (Bianco *et al.* 2006a), the circadian rhythms, the mitotic cycles in early amphibian embryos (Manca, Bianco 2008), a Pseudomonas quorum sensing model (Bianco *et al.* 2006c), the lac operon gene regulatory mechanism in glycolytic pathway (Castellini *et al.* 2009).

Goldbeter's mitotic oscillator was modeled using MP system by Manca, Marchetti (2010a). In this paper it was shown that metabolic P systems yield a robust method for biological modeling. New models based on MP system theory are being developed. For example, recently MP systems were applied to breast cancer research (Bao *et al.* 2017; Bollig-Fischer *et al.* 2014). In the future more biological processes can be expected to be modeled by MP system theory.

The MP system that models the glucose-insulin interactions in the intravenous glucose tolerance test (IVGTT) was proposed by Manca *et al.* (2011). There are several MP models of the IVGTT medical procedure. The simplest example is the following MP system:

$$\begin{aligned} r_1 &: \emptyset \to G, \quad \phi_1 = 0.6\,; \\ r_2 &: G \to \emptyset, \quad \phi_2 = 0.12G + 1.6 \cdot 10^{-6} G^2 I\,; \\ r_3 &: \emptyset \to I, \quad \phi_3 = 49.9 + 0.1G^3\,; \\ r_4 &: I \to \emptyset, \quad \phi_4 = 0.84I\,. \end{aligned} \tag{1.21}$$

IVGTT MP system has two types of parameters. These are rules (r) and regulators (ϕ). The rules describe how the substances of glucose G and insulin I interact within the system. The amounts of these substances increase or decrease depending on which rule is applied. In this case rules r_1 and r_2 increase or decrease the amount of glucose and rules r_3 and r_4 increase or decrease the amount of insulin. On the right side of the equation, the regulators determine the rule application rate. It can be constant like in the case of ϕ_1 or a function of the substance values.

The rules are always the same for all MP systems of IVGTT, but the regulators can depend on the analyzed experimental data set. Therefore the MP system can be adapted for each individual patient as was shown by Manca *et al.* (2011).

As we can see from the applications described in Section 1.3, MP systems have a big variety of different uses. One important and promising application seems to be the IVGTT system that, if implemented in electronic glucose measurement device, could allow to improve diabetes research and treatment.

1.4. Metabolic P System Implementation Ways

Metabolic P systems can be used in a variety of applications, but that requires a way to implement them in some kind of computational device. All implementations so far for the mentioned applications were done in software and run on powerful general purpose computers. However, there are use cases where a smaller, more specialized device may need to be used to perform MP system calculations. This would enable MP systems to be used in applications that require a portable device that performs real-time calculations.

1.4.1. Existing Software Implementations of Metabolic P Systems

There are currently two major implementation ways of MP systems. The first one started as a Psim (Bianco *et al.* 2007). It is a simulation tool developed for MP system modeling. It allows to describe a system by means of MP graphs and simulate

this systems dynamics based on metabolic algorithm. Psim is developed using Java programming language and features a graphic user interface that is used to draw appropriate elements and construct MP graphs. A finished graph can then be simulated and the results can be displayed in a chart form. The Psim software is freely available and can be downloaded[1].

An improved version of the Psim software, called MetaPlab, was introduced by Castellini, Manca (2009). It introduced a new plugin-based architecture which makes the software more versatile and able to perform multiple tasks. In the paper it is described as "virtual laboratory". MetaPlab can have many plugins (Castellini *et al.* 2014) and modifications to help perform the simulation as well as represent the data. There has also been a proposal to represent MP system data in XML format (Manca, Marchetti 2009). MetaPlab software is available under GPL open-source license and can be freely downloaded from MetaPlab web site[2].

Another implementation of MP systems is an open-source Java library appropriately called MpTheory Java Library (Marchetti, Manca 2014). It is also available for download[3]. The provided Java objects can be directly used to model selected MP systems, but the library can also be used within MATLAB™, GNU Octave, Mathematica and R computing environments. This makes the library very versatile and compatible with many systems. As an output it produces a plot of the modeled MP system.

Individual MP systems can also be manually simulated in different programming languages. The mathematical expressions of MP systems can be used to write program code using such computing environments as MATLAB™, shown in Example 1.1.

Example 1.1 (MP system calculation in MATLAB™)

brusselator.m

```
function results = Brusselator(steps)

  % initial values
  xi = 263.4;
  yi = 263.4;

  % state vector
  S = zeros(2,steps);
  S(:,1) = [xi;yi];

  % stoichometric matrix
  M = [1,-2+3,-1,-1;0,-1,1,0];

  % constants
  c1 = 1; c2 = 10^-6; c3 = 0.03; c4 = 0.01;

  for i = 1:steps-1
    F = [c1;c2*S(1,i)^2*S(2,i);c3*S(1,i);c4*S(1,i)];
```

[1] P Systems Software. http://ppage.psystems.eu/uploads/7/7d/Psim.zip

[2] MetaPlab Virtual Laboratory. http://mplab.sci.univr.it/

[3] MpTheory Java Library. http://mptheory.scienze.univr.it/

```
    S(:,i+1) = S(:,i) + M * F;
  end

  results = S;
end
```

Brusselator MP system (Bianco et al. 2006b) is implemented in MATLAB™ *computing environment using the MP grammar shown in Table 1.1.*

1.4.2. Possibilities of Implementation in Hardware

Although MP systems can be implemented using specialized software or relatively simple implementation of MP formulas in MATLAB™, it is also possible to directly implement them in hardware. P systems are known to have been implemented in hardware, but so far there are no known similar implementations of MP systems.

P system implementation in hardware is called Reconfig-P (Nguyen *et al.* 2007). It is a high performance implementation in field-programmable gate arrays. The name of Reconfig-P is based on the reconfigurability feature of FPGA. The implementation can be done by constructing and synthesizing a hardware circuit for a specific P system. Handel-C hardware description language is used for the implementation.

Reconfig-P uses a minimalistic approach in regards to the features of P systems. According to Nguyen *et al.* (2008), only those features of the intuitive conceptual understanding of a P system absolutely necessary to the computational operation of a P system are implemented explicitly as processing units or data structures. This helps to maximize the performance and minimise hardware resource consumption.

An alternative hardware design of Reconfig-P was also proposed by Nguyen *et al.* (2009). It is called the region-oriented design and is intended to promote the understandability of Reconfig-P by more closely reflecting the intuitive conceptual understanding of a P system and promote the extensibility of Reconfig-P by providing a framework within which the future implementation of additional types of P systems can more easily be achieved, and facilitate an elegant region-oriented approach to the distribution and composition of the computational activities occurring in a P system.

MP system specific hardware implementation was theorized by Guiraldelli, Manca (2015b). It was shown that a particular kind of MP systems called positively controlled MP system has the same computational power of a register machine and of a Turing machine, along with a demonstration of the equivalence relation between these models. A register machine is a simple but powerful computational device that is easily translatable to digital computer architecture (Minsky 1967).

The inverse transformation of MP system into register machine was also shown to be possible by Guiraldelli, Manca (2015a). A sofware tool was presented that is capable of transforming either MP systems to register machines or register machines to MP systems. The proof of availability of the bidirectional translation and equivalence

between MP systems and register machines is said to open the way for implementation of circuits based on MP systems. Therefore, hardware implementation is a natural next step. Effective methods and techniques for MP system implementation in hardware need to be developed as so far only theoretical and software implementations are available.

1.5. Quality of Implementation in Hardware

To effectively implement MP systems in hardware, there is a need to estimate the quality of this implementation. There are multiple techniques and criteria that can be used for this. Hardware implementations have their own challenges different from software implementations. Let's start by discussing the techniques that can be used to estimate the quality.

1.5.1. Quality Estimation Techniques

Hardware implementation quality consists of multiple related aspects that need to be evaluated either independently or as part of the system. To evaluate each quality aspect, certain criteria are established that are important for the specific implementation. The collection of these criteria then form the total system quality.

Two techniques can be used to obtain the quality criteria required for evaluation. Most hardware implementation use a specific software provided by the hardware manufacturer. Then the design is implemented using a hardware description language (HDL). Usually there are detailed synthesis reports available in this software. Using various HDL techniques and optimizations, quality of results can be tracked and increased accordingly (Bian *et al.* 2010; Grewal *et al.* 2017; Mametjanov *et al.* 2015).

Because there are many different software measurement and optimization products, an additional technique should be used to increase the reliability of quality measurement. The other way of obtaining the required quality criteria is measuring the physical hardware implementation system. Although this method is slower, it can be used to confirm the reliability of the software estimation by measuring a few selected cases.

1.5.2. Quality Estimation Criteria

The main task of a particular hardware implementation should be considered when estimating its quality. In case of MP system implementation, this task would be the correct and efficient application of arithmetic operations to calculate results based on provided MP formulas. This can be broken down into three groups: calculation accuracy, calculation speed and power consumption of the hardware.

To estimate calculation accuracy, the error of calculation in hardware must be determined. Root mean square error (RMSE) and mean absolute error (MAE) are

the most common ways to evaluate the signal accuracy (Willmott *et al.* 2009). Each metric has its own advantages and disadvantages, although RMSE has proven to be a good indicator of average error (Chai, Draxler 2014). However, in some cases it can be advantageous to take into account both RMSE and MAE (Willmott, Matsuura 2005).

When estimating calculation speed, the meaning of the calculated values should also be considered. The main parameter of hardware implementation speed is the operating frequency that determines how fast the calculations are performed. However, depending on the calculated values, there can be a number of calculation cycles required to get any meaningful result (instead of an intermediate value). In the case of MP systems, when estimating calculation speed, a single iteration of substance values at discrete time n can be considered a meaningful value. Therefore when estimating calculation speed, the frequency should be divided by the number of cycles required to get the meaningful value.

Power consumption greatly depends on the architecture of used hardware and implementation specifics (Jamieson *et al.* 2009). Different designs use a varying number and type of hardware elements. Power consumption is contrary to the other criteria of accuracy and speed. Therefore each implementation can be considered a compromise between power, speed and accuracy – when one quality criteria is increased, other criteria inevitably decrease.

1.5.3. Field Programmable Gate Array Specificity

Reconfigurable hardware such as Field Programmable Gate Array (FPGA) is an integrated circuit that allows to convert the circuit board level into a chip level product after manufacturing. FPGA are widely used in communications (Stašionis, Serackis 2013, 2016), medicine (Senthilkumar, Umamaheswari 2012), high performance computing, industrial control with high speed, low power consumption, high reliability, high density (Sun *et al.* 2002). FPGA gives an opportunity to process the incoming data with necessary speed and without the complexity of an application-specific circuit (Haselman *et al.* 2009) thereby reducing power consumption and improving reliability. Depending on the manufacturer and chip model, each FPGA has a finite number of spatial resources available. There are three main types of FPGA resources that can limit the possible implementation size: arithmetic, logic and memory. Each of these resources need to be taken into account when estimating implementation quality.

Arithmetic resources in FPGA are made of Digital Signal Processor (DSP) slices. They are optimized to be able to perform fast arithmetic addition and multiplication operations. Depending on the design of the implementation, one or more reconfigurable DSP slices can be used. They are the most efficient when operating with integer or fixed point numbers.

Logic resources in FPGA are often limited by the number of Look-Up tables (LUT) available. This depends on the manufacturer and is provided in the FPGA specification. These logic resources can be used to create configurable logic blocks that perform various logic functions. Any implementation needs to perform at least some

logic operations, therefore logic resource consumption should always be estimated when determining implementation quality.

Memory resources in FPGA can be synthesized using specific slices consisting of LUT elements. Block Random Access Memory (BRAM) is synthesized to create buffers for data storage. The kind of data stored in memory depends on the implementation. In some designs there may be no need to use dedicated memory blocks. In such case memory usage is not counted when estimating implementation quality.

Most FPGA implementations require to output some data, either to a human readable display or to other electronic instruments. This way an interface is formed between the FPGA and the external environment. The maximum width of this interface is limited by the number of Input-Output Blocks (IOB) available on the FPGA. When more IOB is available, a higher precision (quality) signal can pass over the interface.

Table 1.2. Comparison of different field programmable gate array chip families (Intel 2018; Xilinx 2018)

Manufacturer	Chip Family	Maximum available resources		
		Logic elements	DSP elements	Memory, Kb
Xilinx	Virtex-7	1,139,200	3,600	67,680
	Kintex-7	477,760	1,920	34,380
	Artix-7	33,650	740	13,140
	Spartan-7	16,000	160	4,320
Intel	Stratix 10	5,510,000	5,760	229,000
	Arria 10	1,150,000	1,688	53,000
	Cyclone 10	220,000	192	11,740
	MAX 10	50,000	144	1,638

There are two main FPGA manufacturers – Xilinx and Intel (formerly Altera) – that together control almost 90 % of the market (Dillien 2017). The latest FPGA chip families are the 7 series of Xilinx and the 10 series of Intel. Comparison of these chip families is shown in Table 1.2. The capability spectrum of these chip families shows that there are many possibilities to choose an appropriate FPGA according to the needed specification. There are chips for both simple cost effective and complex state of the art applications.

1.6. Conclusions of the First Chapter and Formulation of the Thesis Objectives

After reviewing Metabolic P systems, their applications and their implementation possibilities, the following conclusions can be made and the tasks of the thesis can be defined:

1. Metabolic P systems are an extension of membrane computing, also known as P systems. Membrane computing, including MP systems, is a part of the research field called natural computing.
2. A set of functions called MP grammar is used to calculate Metabolic P systems. MP grammar and the set of initial system states can be defined by approximating real periodical functions.
3. Metabolic P systems have applications in mathematics, biology, chemistry and other fields. Using MP grammar and initial states provided in publications of these implementations, a set of bioengineering data suitable for testing and validating Metabolic P system must be created.
4. There are various different software implementations of membrane computing and specifically Metabolic P systems. Hardware implementation of a few P system types was attempted and shown to be possible, but there are no known hardware implementations of Metabolic P systems. By using theoretical results of Metabolic P systems and knowledge of other practices, novel solutions and methodics to transform Metabolic P system concepts to FPGA structural elements and signal processing schemas should be offered.
5. Hardware implementations in FPGA systems can be evaluated by comparing calculation accuracy, speed, resource usage, power consumption and interface complexity metrics. The quality characteristics specific to the Metabolic P system transformation to FPGA should be revealed.
6. To confirm the quality evaluation criteria measurements a real time metabolic process imitation and testing system should be created and researched. For this task Field Programmable Gate Arrays are suitable hardware devices.

2

Quality Evaluation Criteria of Metabolic P System Implementation in Hardware

Before trying to implement Metabolic P system in hardware, evaluation criteria of such implementation must be established. When evaluating MP system implementations it is important to define their quality. The overall quality is not a single measurement or parameter. It is a combination of different factors that affect the implementation in a specific way, as was shown in Section 1.5. That means that usually the implementation has advantages according to some metrics and disadvantages according to another metrics.

The general structure of implementation quality criteria specific to MP systems is introduced in Section 2.1. Quality criteria components specific to calculation of MP systems are discussed in Section 2.2. The calculation specific quality criteria depend on the specifics of MP systems. At the same time, the implementation specific quality criteria, presented in Section 2.3, depend in the most part on the selected implementation methods and techniques. Likewise, the environment specific criteria mostly depend on the selected FPGA model and are presented in Section 2.4. Finally, the ways of evaluating the complete MP system implementation quality in FPGA are discussed in Section 2.5. An example calculation as well as graphical representation are presented to illustrate the complete quality evaluation.

The research results are published in author publications (Kulakovskis 2015; Kulakovskis, Navakauskas 2015, 2016). The main results are announced in international AIEEE (Riga, 2015) and national "Science – Future of Lithuania" (Vilnius, 2015, 2016) scientific conferences.

2.1. Structuring of Quality Evaluation Criteria

Although the most important implementation quality evaluation metrics were discussed in detail in Section 1.5, it is important to be able to determine total implementation quality. It is needed to objectively compare different implementations and to be able to determine how the overall quality of the implementation is affected by any single metric.

2.1.1. State of the Art

In HariKumar *et al.* (2012) an FPGA based fuzzy proportional integral derivative controller for insulin pumps is presented. The controller uses fuzzy logic to manage the movement of the infusion pump. For the measurement of glucose levels in the patient's blood a non-invasive Photoglucometer sensor is used. To represent the dynamics of the glucose and insulin concentrations, two models by Ackerman *et al.* (1965) are used.

A fuzzy logic based FPGA controller was also used in Gharghory, El-Dib (2016). The presented closed loop control system is called an artificial pancreas. Glucose and insulin concentration dynamics are modelled using Bergman's minimal model. This model has a minimal number of parameters and represents only the essential dynamics of the glucose and insulin system, but is suitable for real-time implementation. A fuzzy logic control system to control the blood glucose level continuously, based on current and previous values, was designed in VHDL.

An FPGA based blood glucose level control system called an artificial pancreas is also developed in Ghorbani, Bogdan (2013). The glucose and insulin concentration dynamics are modelled using a novel mathematical model based on fractional calculus concepts.

A glucose–insulin feedback system based on FPGA is developed in Dutta, Botros (2013). It uses the mathematical model proposed by Tolic *et al.* (2000). The calculations are performed on FPGA using the Digital Differential Analyzer algorithm. The algorithm is used in order to solve the mathematical model of insulin–glucose dynamics and to write the HDL.

A real-time FPGA based device to regulate insulin delivery was proposed in Nguyen *et al.* (2013). Pancreatic β-cells were used as a biosensor to monitor the glucose concentration in the blood. The reliability of measurements and patient security were important problems because insulin overdose can cause hypoglycaemia and threaten the lives of patients. Therefore, parallel measurements in multiple channels were made. An FPGA based embedded device is proposed for parallel signal processing.

The issues of implementing complex differential equations in hardware were examined in Li *et al.* (2015). Cobelli's glucose–insulin model was implemented using Simulink blocks with the DSP Builder. To overcome FPGA implementation issues, a simplification of Cobelli's glucose-insulin model was proposed using model order

reduction methods. The implementation accuracy was evaluated according to instantaneous and RMSE of the predicted glucose changes.

A recurrent high-order neural network, trained with an extended Kalman filter is used to obtain a neural model to regulate the glucose level for type 1 diabetes mellitus patients Romero-Aragon *et al.* (2014). The neural inverse optimal controller uses a control Lyapunov function to obtain an inverse optimal control law that calculates the insulin delivery rate. The neural model is implemented in FPGA using Verilog and tested by monitoring a virtual patient simulated by a computer. The calculation accuracy is evaluated using RMSE and standard deviation of the identification and tracking.

The importance of fault tolerance in the design of the artificial pancreas is highlighted in Vavouras *et al.* (2016). Dual Modular Redundancy techniques are used in a hybrid ASIC/FPGA system to detect and correct transient and permanent faults. The reconfiguration feature of FPGA is mentioned as a way of correcting faults. Partial reconfiguration and bitstream relocation techniques are used for fault recovery. The system was implemented using VHDL and evaluated as a medical device according to International Electrotechnical Commission safety standards. Hardware dependability, timing dependability and application related metrics were the criteria used for the evaluation.

Glucose regulation is used as a case study in Vouzis *et al.* (2009). The glucose dynamics are modelled using Bergman's minimal model and implemented in FPGA using a novel hardware architecture for embedded real-time model predictive control. In this case study the possibilities of reduced power consumption are discussed. As the frequency of the glucose sensor is in the order of minutes, the operating frequency and voltage of the FPGA based system can be reduced. This could potentially result in a major reduction of the system power consumption.

Most of the above-mentioned implementation techniques are evaluated by comparing the FPGA resource usage, speed, power consumption, reliability and other parameters discussed in Section 1.5. These FPGA implementations of various glucose-insulin monitoring systems use a variety of different quality criteria, but none of them are specific to MP systems. Also there is no clear consensus on what quality criteria are the most important and should be used in an experimental evaluation. Therefore, in the next section let's analyze the quality criteria that are useful for MP system implementation in FPGA.

2.1.2. Combined Quality Metrics

After investigating general hardware implementation quality in Section 1.5 and other recent hardware implementations in the previous section, the most suitable quality criteria specifically for MP system implementation in FPGA were selected. By analyzing other implementation examples and having in mind the specifics of MP systems, five different criteria were determined that are required to determine complete implementation quality.

Quality of MP system FPGA implementation can be represented by a set of these five criteria, each normalized to vary in a range from 0 to 100:

$$Q_{\text{MP}} = \{Q_{\text{A}}, Q_{\text{T}}, Q_{\text{R}}, Q_{\text{P}}, Q_{\text{I}}\} \ , \tag{2.1}$$

where Q_{MP} – the combined quality of MP system implementation; Q_{A} – the quality of calculation accuracy; Q_{T} – the quality of result generation throughput; Q_{R} – the quality of FPGA resource usage; Q_{P} – the quality of FPGA power consumption; Q_{I} – the quality of FPGA interface complexity.

Quality of accuracy Q_{A} represents the inverse of error of the results when compared to the reference point. The magnitude of calculated MP system substance values, and therefore the calculation error, differ depending on the selected MP system.

Additionally, if there are multiple substances in a single MP system, the magnitude of their values can also differ. Therefore to approapriatelly compare errors of different substances and different MP systems, normalized errors need to be used. Normalized error is also used to adjust the error percentage to be within the range of 1 to 100 like every other quality criteria.

The actual error value can be calculated using different techniques. MP system substance output values are represented by a signal vector when MP system has only one substance and a signal matrix when MP system has multiple substances. Therefore MP system implementation accuracy calculation:

$$Q_{\text{A}} = (1 - E) \times 100 \ [\%] \, , \tag{2.2}$$

with E expressed by normalized root mean square error

$$E_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(\widehat{x}(n) - x(n)\right)^2} \Bigg/ \left(x^{\top} - x^{\perp}\right) \ , \tag{2.3}$$

or by normalized mean absolute error

$$E_{\text{MAE}} = \frac{1}{N} \sum_{n=1}^{N} |\widehat{x}(n) - x(n)| \Bigg/ \left(x_{\top} - x_{\perp}\right) \ , \tag{2.4}$$

where $x(n)$ – reference MP system substance value at discrete time n; $\widehat{x}(n)$ – approximated MP system substance value at discrete time n; $x_{\top}$ and $x_{\perp}$ – maximum and minimum substance values; N – total length of the MP system substance value approximation results vector. In case there are multiple MP system substances and results matrix is used, the error values must also be averaged over each results matrix row.

RMSE usage means that extreme points of the substance value errors have less impact on the total error value when compared to MAE usage. In most MP system

calculation cases this is preferred but it depends on the exact use case and the implemented MP system. MAE use is preferred where error of a single substance value needs to be emphasized.

Quality of throughput Q_{T} represents the percent of how close the MP system substance value generation frequency is to the maximum clock speed:

$$Q_{\mathrm{T}} = \frac{f_{\mathrm{val}}}{f_{\mathrm{clock}}} \times 100\ [\%]\,, \tag{2.5}$$

where f_{val} – MP system substance value generation frequency (Hz); f_{clock} – maximum possible value generation frequency of the selected FPGA chip (Hz), that is closely related to the used clock speed that can not be exceeded and is provided in data sheets of FPGA chips (usually in the range of 100 to 300 MHz).

The MP system substance value generation speed f_{val} indicates how frequently the values are obtained. f_{val} stays the same even if there are multiple substances in an MP system. This is because of the specifics of MP systems where to calculate the next substance value, only the values of previous calculation cycles are needed. This means that advantage of FPGA architecture can be used to calculate substance values simultaneously.

Quality of resource Q_{R} usage represents the percentage of used FPGA resources. There are three different types of FPGA spatial resources: LUT, DSP and BRAM. To compare the total resource usage, they need to be converted to an equivalent parameter. For this purpose the total used resources are converted to LUT equivalent units:

$$Q_{\mathrm{R}} = \frac{N_{\mathrm{maxLUT}} - N_{\mathrm{eqLUT}}}{N_{\mathrm{maxLUT}}} \times 100\ [\%]\,, \tag{2.6}$$

with total used resources, expressed in LUT equivalent units

$$N_{\mathrm{eqLUT}} \equiv w_{\mathrm{LUT}} N_{\mathrm{LUT}} + w_{\mathrm{DSP}} N_{\mathrm{DSP}} + w_{\mathrm{BRAM}} N_{\mathrm{BRAM}}\,, \tag{2.7}$$

where N_{LUT} – number of used logic resources; N_{DSP} – number of used arithmetic resources; N_{BRAM} – number of used memory resources; N_{maxLUT} – total number of LUTs available; w – weights (coefficients) that determine the relation with LUT equivalent resources.

The weights used for converting of FPGA resources to N_{eqLUT} depend on the architecture of used FPGA chip. They can be determined by consulting specifications of the FPGA chip model[1]. For example, for Xilinx 7th series FPGA chip family $w_{\mathrm{DSP}} = 196$ because 196 LUT need to be used to synthesize one DSP. Similarly, $w_{\mathrm{BRAM}} = 576$ because one LUT stores 32 b of memory when configured as a memory element and one BRAM stores 18432 b.

[1] Xilinx Inc., 7 Series FPGAs Overview (ver2.6) (28 February 2018). https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

Additionally, $w_{\text{LUT}} = [0.7, 0.8]$ because FPGA have limited routing resources and not all LUT can be utilized at the same time. When all aspects of resource usage are converted to the same dimensions, they can be combined using addition to get the final number of LUT equivalent resources.

Quality of power consumption Q_{P} represents the percentage of consumed power by the FPGA implementation design in relation to the maximum available power that is limited by FPGA thermal and power design:

$$Q_{\text{P}} = \frac{P_{\text{TOCP}}}{P_{\text{TDP}}} \times 100\ [\%]\,, \tag{2.8}$$

where P_{TOCP} – Total On-Chip Power that can be estimated either be physically measuring the consumed power or by using special tools provided by FPGA manufacturers, such as Xilinx XPower Analyzer; P_{TDP} – Thermal Design Power that is the maximum possible power consumption according to the FPGA design, provided in FPGA data sheets.

The power consumption can also vary depending on the environmental conditions such as ambient temperature. Therefore it is important to maintain the same conditions between measurements or take the changed conditions into account when estimating the quality of power consumption.

Quality of interface complexity Q_{I} represents the percentage of used FPGA input-output block resources. They are used for transferring the initial calculation data to the chip and the final calculation result to the external environment. The available number of IOBs is determined by the used FPGA model:

$$Q_{\text{I}} = \frac{N_{\text{IOB}}}{N_{\text{maxIOB}}} \times 100\ [\%]\,, \tag{2.9}$$

where N_{IOB} – number of used input-output resources that depend on the number of output signals and their width; N_{maxIOB} – total number of IOB available according to the used FPGA chip, usually in the range of 100 to 1000 blocks.

2.2. Calculation Specific Criteria

There are quality evaluation criteria that are dependent only on the calculations of MP systems. These criteria can be theoretically calculated without using a specific implementation hardware.

2.2.1. Calculation of Metabolic P Systems

Unlike many methods used for modeling of similar objects, MP systems do not use differential equations and in principle can be efficiently implemented in hardware by

using only basic mathematical operations. As an example, an imaginary MP system described by the following MP graph is presented:

$$\begin{aligned} r_1 &: 2A \to 3B, \quad \phi_1 = 5A + 8.5; \\ r_2 &: B \to A, \qquad \phi_2 = A + B - 1\,. \end{aligned} \tag{2.10}$$

This equation consists of two parts: left, and right. The left part of the equation represents the reactions r, and the left part represents regulators ϕ that determine the rate at which the reaction is happening. The MP system has two rule and regulator pairs (r_1 and ϕ_1, r_2 and ϕ_2), two substances (A and B), and three constants used in the regulators.

Substances are the main variables in MP systems. In this case, there are two substances: A, and B. These substances must have an initial amount that will be used at the start of MP system calculation.

Rules are the main MP system reactions that transform one or more substances into each other, introduce substances from the environment or expel them to the environment. In this case, there are two reactions. The first one transforms an amount of two of substance A into an amount of three of substance B. In other words, $2A$ is subtracted and $3B$ is added.

Constants are the numerical values that do not change during the MP system iterative calculation process. Constants participate in the addition, subtraction or multiplication operations of MP system regulators. In this case, there are three constants represented by two integer numbers and one fractional number.

Regulators, sometimes also called fluxes, determine the rate at which the reactions are happening. Each MP system rule has its own regulator expression that contains mathematical operations using the previously defined constants and substances. The calculations example of substance values is:

$$\begin{aligned} A(n) &= \phi_2 - 2\phi_1 = B(n-1) - 9A(n-1) - 18, \\ B(n) &= 3\phi_1 - \phi_2 = 14A(n-1) - B(n-1) + 26.5\,. \end{aligned} \tag{2.11}$$

The values of each substance at time n depend on previous values of other substances. This is indicative of most MP systems. This feature of MP systems means that even small errors would be amplified on each subsequent calculation step. Therefore error measurement and mitigation is an important aspect of MP system calculation.

2.2.2. Accuracy Evaluation Metrics

MP systems consist of a number of substances that is greater or equal to one. Therefore accuracy calculation needs to take this into account. All substances are equally important to be modeled accurately. For this reason the normalized root mean square error is averaged across all substances.

The quality of accuracy of MP systems with multiple substances is:

$$Q_{\mathrm{A}}^{\mathrm{MP}}\left(N_{\mathrm{WLI}}, N_{\mathrm{WLF}}\right) \triangleq 100 - \frac{1}{|X|}\sum_{x\in X} E_{\mathrm{RMSE}}^{(x)}\left(N_{\mathrm{WLI}}, N_{\mathrm{WLF}}\right), \tag{2.12}$$

with normalized root mean square error of substance x represented by

$$E_{\mathrm{RMSE}}^{(x)}\left(N_{\mathrm{WLI}}, N_{\mathrm{WLF}}\right) = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\widehat{x}\left(n; N_{\mathrm{WLI}}, N_{\mathrm{WLF}}\right) - x(n)\right)^2}}{\left(x^{\top} - x^{\perp}\right)}, \tag{2.13}$$

where $x(n)$ – reference substance value at discrete time n; $\widehat{x}(n)$ – approximated substance value at discrete time n; X – set of substances of a particular MP system; N – total length of the MP system substance value approximation results vector; N_{WLI} – number of bits used for integer part of fixed point arithmetic; N_{WLF} – number of bits used for fractional part of fixed point arithmetic.

2.2.3. Impact of Using Fixed Point Calculations

The MATLAB™ computing environment can be used to determine the impact of using fixed point arithmetic to the accuracy of MP system modeling results. These results can be applied before implementing MP systems in dedicated hardware platforms.

For the purpose of demonstration, the Brusselator MP system, described in Section 1.3.2, was implemented using fixed and floating point calculations. It was found that for fixed point calculations to work, at least 32 b precision is needed. As shown in Fig. 2.1, when using 16 b precision, modeling results in constant value throughout the timeline. When experimenting with different bit counts for the fractional part, it was found that the best results for this particular model are achieved when assigning 20 bits out of 32 for the fractional part. Therefore, 20 bits for the fractional part should be used.

Fig. 2.2 shows the results when using 32 b precision. In the first period of oscillation the difference between floating and fixed point is minimal, but it is increased in each following period. This results in the loss of precision, because each following point in time is:

$$x(n+1) = x(n) + \Delta n\,, \tag{2.14}$$

where the new $x(n+1)$ value always depends on the previous one value $x(n)$ and Δn is a product of a stoichiometric matrix and a flux vector. In the Brusselator model, they are calculated from MP grammar previously shown in Table 1.1.

The precision of the Brusselator MP system modeled using 32 b fixed point calculations decreasing over time (Fig. 2.3). At times the inaccuracy can reach more than 400 points – close to the total amplitude of the oscillation. It is caused by the specifics of the modeled system, as the oscillation includes some sharp value changes.

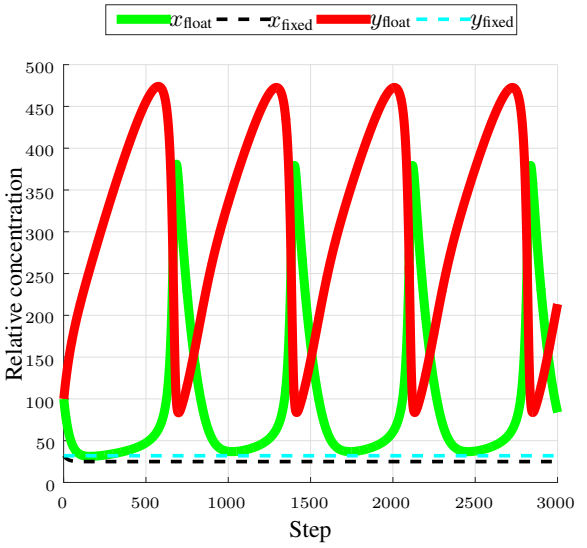


Fig. 2.1. Signals of the Brusselator modeled using floating point and 16 bit fixed point calculations

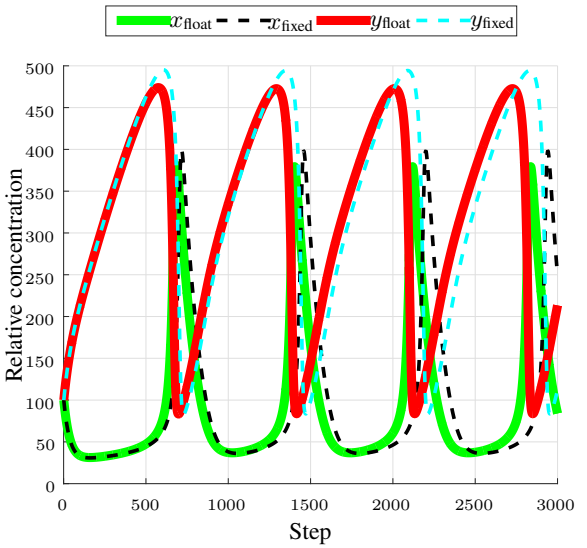


Fig. 2.2. Signals of Brusselator modeled using floating point and 32 bit fixed point calculations

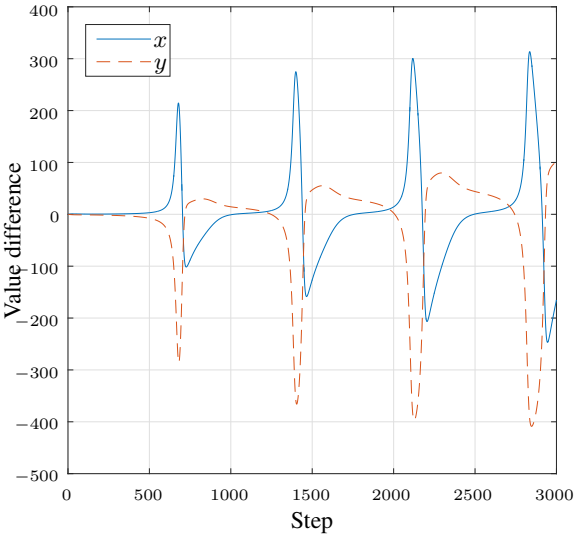


Fig. 2.3. x and y value difference when calculated using floating point and fixed point calculations

Table 2.1. Modified Metabolic P grammar of the Brusselator reaction

Reactions	Regulators
$r_1 : \emptyset \rightarrow 0.01x\,;$	$\phi_1 = 100\,;$
$r_2 : 0.02x + 0.01y \rightarrow 0.03x\,;$	$\phi_2 = 0.01x^2 \cdot 0.01y\,;$
$r_3 : 0.01x \rightarrow 0.01y\,;$	$\phi_3 = 3x\,;$
$r_4 : 0.01x \rightarrow \emptyset\,.$	$\phi_4 = x\,.$

To increase the accuracy of fixed point calculations, an adjustment to MP grammar of the Brusselator reaction can be done by using a scale factor (SF). The main purpose of using SF is to avoid having numbers that require more bits to be represented accurately in digital systems. The resulting MP grammar with scaled numbers is shown in Table 2.1. By applying SF to the calculations, an improved fixed point calculation accuracy can be achieved. The modeling results, as shown in Fig. 2.4, are very close between floating and fixed point systems. Therefore, it can be concluded that the modification of MP grammar using a SF is an effective way of improving fixed point calculation accuracy. The basis of this modification is a decreased amplitude of constants in MP grammar. This is achieved by dividing all constants in the reactions side of the table and multiplying the constants in the regulators side of the table by the same SF. As long as the multiplier and divisor are the same, the resulting MP grammar is equivalent to the original.

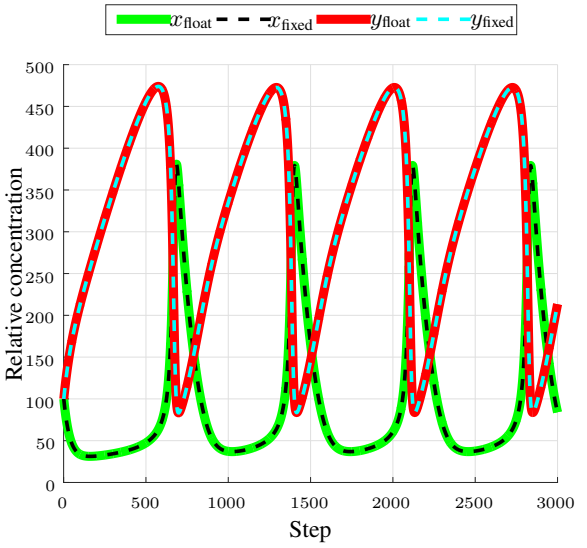


Fig. 2.4. Signals of the Brusselator modeled using floating point and 32 bit fixed point calculations using modified Metabolic P grammar

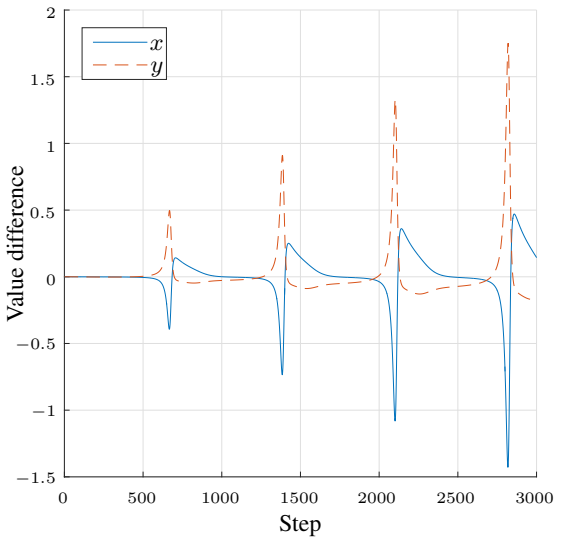


Fig. 2.5. x and y value difference calculated using floating point and fixed point calculations and using modified Metabolic P grammar

When using modified MP grammar the difference, shown in Fig. 2.5, between floating and fixed point calculations is more than 250 times lower. This results in greatly increased accuracy and prevents the oscillation from amplifying the error.

The shown MP grammar SF modification method can be applied to most other MP systems. The main rule when adjusting the formula is to multiply regulators and divide reactions by the same amount. Of course, like in any other limited precision calculations, it is beneficial to avoid multiplying very small or very large numbers. In this modeling it was avoided when calculating ϕ_2 by adjusting the order of multiplied values. In case of FPGA implementation, shift operations can be an effective way of modifying the coefficients to achieve this improvement in accuracy. In this case an SF in the order of 2^n should be used to effectively take advantage of shift operations.

2.3. Implementation Specific Criteria

The number of used FPGA resources as well as design frequency are provided by Synthesis Report when Xilinx hardware is used (other manufacturers provide equivalent software tools). Calculation speed can be determined from the Timing Summary section of the report. It is measured using maximum design frequency value in MHz or minimum period value in ns. The number of used FPGA resources is provided in Device Utilization Summary section of the Synthesis Report and is measured by providing the total number of cells used and free.

2.3.1. Calculation Speed

MP systems can be calculated using different methods. Some of these methods can use multiple parallel channels to speed up the calculation. Other methods can rely on performing calculations in batches during multiple clock cycles. Both of these methods need to be taken into account when estimating MP system value generation speed (Hz):

$$f_{\text{val}} \triangleq \frac{f^{\top}}{\tau}, \tag{2.15}$$

where $f^{\top}$ – maximum clock frequency after place and route stage (Hz); τ – latency of one calculation (number of cycles needed to get the final result) that depends on the number of operations needed to obtain MP system substance values and varies according to the complexity of selected MP system.

When value generation speed is defined, the quality of MP system throughput can be written as:

$$Q_{\text{T}}^{\text{MP}} \triangleq \frac{f^{\top}/\tau}{f_{\text{clock}}} \times 100\ [\%]\,. \tag{2.16}$$

2.3.2. Usage of Field Programmable Gate Array Spacial Resources

For MP system calculation, memory (BRAM) resources are not synthesized, as they are not required according to the specifics of the MP system use case. Therefore w_{BRAM} always equals zero. The resource usage of MP system implementation in FPGA depends on the number of synthesized LUT and DSP elements and the (2.7) equation becomes:

$$N_{\text{eqLUT}} \triangleq w_{\text{LUT}} N_{\text{LUT}} + w_{\text{DSP}} N_{\text{DSP}}\,, \tag{2.17}$$

where N_{LUT} – number of used logic resources; N_{DSP} – number of used arithmethic resources; N_{eqLUT} – total number of used FPGA resources, expressed in LUT equivalent units; N_{maxLUT} – total number of LUTs available (depending on the exact FPGA model used).

The w_{LUT} and w_{DSP} coefficients used in (2.17) are chosen according to FPGA model used for MP system implementation. They depend on the specific architecture of the FPGA chip family. An example of the coefficients is provided in Section 2.1.2. Using the described definitions, the quality of MP system resource usage is written as follows:

$$Q_{\text{R}}^{\text{MP}} \triangleq \frac{N_{\text{maxLUT}} - w_{\text{LUT}} N_{\text{LUT}} - w_{\text{DSP}} N_{\text{DSP}}}{N_{\text{maxLUT}}} \times 100\ [\%]\,. \tag{2.18}$$

2.4. Environment Specific Criteria

When implementing MP systems in FPGA, the relationship of the implementation with the external environment should be taken into account. Two main aspects of this relationship are power consumption and input-output capabilities. These aspects can influence the total quality of the implementation and need to be assessed with separate criteria. These criteria are the most important to investigate when choosing which FPGA chip model should be used, as well as further circuit board design.

2.4.1. Power Consumption

Power consumption of the FPGA implementation depends on the synthesized schematics of the MP system and the exact model of the FPGA chip. It is also closely related to the thermal characteristics of the implementation, that affect the total consumed power.

The TOCP can be estimated using special tools provided by FPGA manufacturer. In case when Xilinx FPGA chips are used, TOCP can be precisely estimated using Xilinx XPower Analyzer tool. The parameters used for the estimation such the resource consumption of the implementation, maximum frequency and the used FPGA model are obtained automatically by the tool. When considering the parameters that have an

effect on the power consumption of MP systems implemented in FPGA, its quality can be described as the following expression:

$$Q_{\mathrm{P}}^{\mathrm{MP}}(T_{\mathrm{amb}}, N_{\mathrm{LUT}}, N_{\mathrm{DSP}}, f^{\top}) \triangleq \frac{P_{\mathrm{TOCP}}(T_{\mathrm{amb}}, N_{\mathrm{LUT}}, N_{\mathrm{DSP}}, f^{\top})}{P_{\mathrm{TDP}}} \times 100\ [\%], \quad (2.19)$$

where T_{amb} – ambient temperature (should be the same for all compared implementations); N_{LUT}, N_{DSP} – used FPGA resources; $f^{\top}$ – maximum clock frequency.

2.4.2. Interface Complexity

The implementation of MP systems requires some data to be transfered to and from the FPGA. Depending on the implementation, the input to the FPGA can consist of the initial values of substances, constants used in MP systems or various selectors. The output is the calculated values of substances in discrete time intervals.

For these input-output operations to be available, some number of IOBs is used when synthesizing the implementation. In case of MP system implementation, the amount of used input-output resources depends on the accuracy needed (bigger binary word length uses more resources) and the number of substances in the MP system:

$$Q_{\mathrm{I}}^{\mathrm{MP}}(N_{\mathrm{WL}}, |X|) \triangleq \frac{N_{\mathrm{IOB}}(N_{\mathrm{WL}}, |X|)}{N_{\mathrm{maxIOB}}} \times 100\ [\%], \quad (2.20)$$

where N_{WL} – binary word length; $|X|$ – number of substances used in the implemented MP system. These parameters determine the number of IOB needed for implementing the selected MP with the selected word length. It is an important criteria when choosing what FPGA chip to use as it must have an appropriate number of IOB to be able to output the calculated signal values.

2.5. Evaluation of Complete Implementation Quality

When MP system is implemented in hardware, it must be validated to confirm that it is working as intended. The evaluation must determine the accuracy of calculations performed in FPGA. To determine the accuracy, some point of reference is needed.

2.5.1. Reference Implementation in Software

MATLAB™ software can be used to determine the accuracy of MP system calculations. The implementation in software can be used as a reference point as it uses floating point arithmetic that is more precise than that available in fixed point FPGA implementation. Also, calculations in software do not have many limitations of the same calculations performed in hardware, such as space and timing constraints.

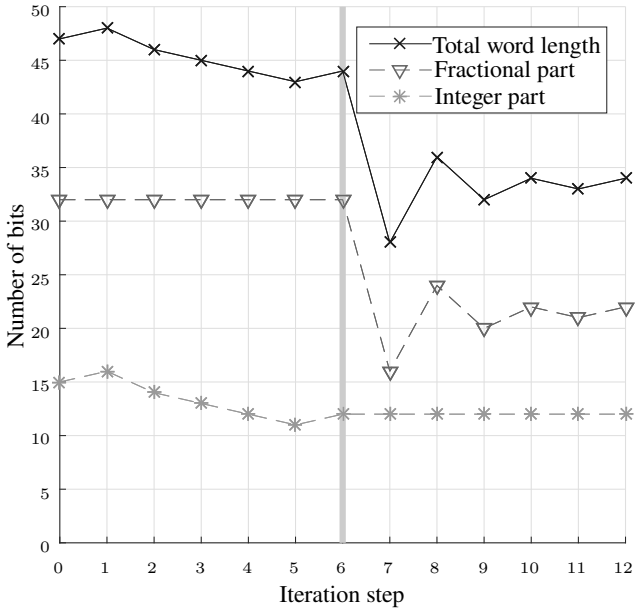


Fig. 2.6. Process of finding the number of bits for the implementation with the maximum error of 1 %. The gray bar emphasizes the change of optimization target – from the integer to the fractional part

The impact of fixed point arithmetic can also be simulated using an algorithm that compares the results with floating point implementation (Kulakovskis, Navakauskas 2016). First, Algorithm 2.1 tries to determine the required number of integer bits. As any decimal integer number can be represented in binary, there is a point where increasing the number of bits does not yield any improvement in the accuracy of the calculation.

When the initial number of bits is selected, a system is modeled using that number of bits, and the error is calculated. Then the same operation is repeated using the integer word length greater by one. When the errors are compared, the algorithm determines whether the error is decreasing when the integer part of the word length is increasing. If it is true, this means that accuracy can still be improved by increasing the number of integer bits. If it is not true, the algorithm tries to lower the number of integer bits to optimize the system.

When the optimal number of integer bits is found, the second Algorithm 2.2 that tries to find the optimal number of fractional bits is started. Initially, a number of fractional bits that is a factor of two is selected, and then the binary search algorithm is used to get the number of fractional bits that yield the closest error value to the requested target.

Algorithm 2.1 (Selection of integer word length)

Let MP_{float} – reference implementation.
Select initial integer and fractional word lengths: $WL_{\text{I}}, WL_{\text{F}}$.
$E_1 = \text{get\_error}(MP(WL_{\text{I}}, WL_{\text{F}}), MP_{\text{float}})$
$WL_{\text{I}} = WL_{\text{I}} + 1$
$E_2 = \text{get\_error}(MP(WL_{\text{I}}, WL_{\text{F}}), MP_{\text{float}})$
if $E_2 >= E_1$ **then**
  $E_2 = E_1$
  while $E_2 <= E_1$ **do**
    $WL_{\text{I}} = WL_{\text{I}} - 1$
    $E_2 = \text{get\_error}(MP(WL_{\text{I}} - 1, WL_{\text{F}}), MP_{\text{float}})$
  end while
else
  $WL_{\text{I}} = WL_{\text{I}} + 1$
  $E_1 = \text{get\_error}(MP(WL_{\text{I}}, WL_{\text{F}}), MP_{\text{float}})$
  while $E_1 < E_2$ **do**
    $WL_{\text{I}} = WL_{\text{I}} + 1$
    $E_2 = E_1$
    $E_1 = \text{get\_error}(MP(WL_{\text{I}}, WL_{\text{F}}), MP_{\text{float}})$
  end while
  $WL_{\text{I}} = WL_{\text{I}} - 1$
end if
return WL_{I}

IVGTT MP system, described in Section 1.3.3, was modeled in software using floating and fixed point arithmetic. The previously described algorithm was used to demonstrate the use of reference implemention in finding the optimal binary word length.

Algorithm 2.2 (Selection of fractional word length)

Let E_{T} – target error, MP_{float} – reference implementation,
WL_{I} – previously optimized integer word length.
Select initial fractional word length: WL_{F}.
Initial adjustment step $s = WL_{\text{F}}$ and step divisor $d = 1$
while $s > 1$ **do**
  $E = \text{get\_error}(MP(WL_{\text{I}}, WL_{\text{F}}), MP_{\text{float}})$

```
    if E >= E_T then
        s = floor(s/d)
        WL_F = WL_F + s
    else
        if WL_F − 1 already tried then
            return WL_F
        end if
        d = 2
        s = floor(s/d)
        WL_F = WL_F − s
    end if
end while
return WL_F
```

The target error was set to be not more than $1\,\%$. To achieve this accuracy, the optimal word length finding algorithm was started. For the IVGTT MP system, 12 b were needed for the integer part. After finding the optimal integer word length, a binary search algorithm was started to find the optimal number of bits for the fractional part. The process of finding the optimal number of bits is illustrated in Fig. 2.6. The algorithm selected 22 b as the optimal number of bits for the fractional part. This means that in total, the word length of 34 b plus the sign bit was used.

Twelve steps were required to find the optimal word length for the IVGTT MP system. The number of steps for determination of the integer and fractional parts is almost equal. Although this depends on the "guess" of the initial values, they were not specifically tailored to suit the selected MP systems. A starting point of a total of 48 b (15 b selected for the integer part, 32 b – for the fractional part, and 1 for the sign) was selected, which mirrors the capabilities of DSP cells of most FPGA.

It should be noted that the developed algorithm only searches for the minimum number of bits that achieve the requested accuracy. It does not, however, check whether the selected binary word length is effective when MP system is implemented in FPGA. It assumes that the use of x b is always better than the use of $(x + 1)$ b in terms of system performance, and always worse in terms of system accuracy.

2.5.2. Comparing Different Implementations

The implementation quality of Metabolic P systems consists of five components as shown in (2.1). The total quality Q_{MP} is a set of different separate qualities Q that represent a particular aspect of MP system implementation quality.

Each aspect of implementation quality Q can be more or less important depending on the needs of a particular FPGA implementation project. Therefore often there is a need to easily compare these quality aspects in a visual way. One example of such

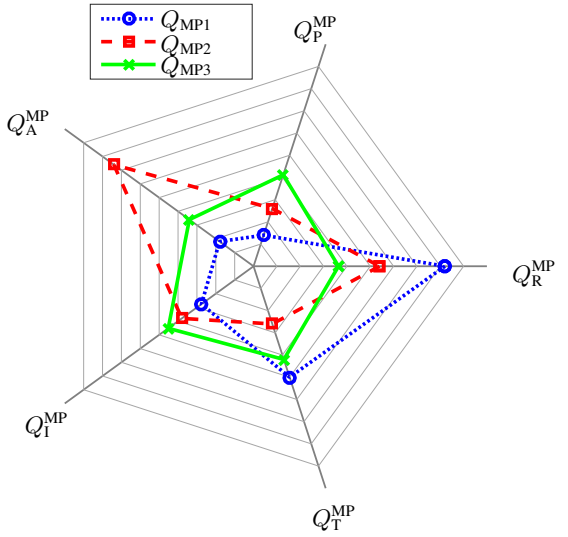


Fig. 2.7. Comparison of three different metabolic P system implementation qualities, sorted by Q_{MP1}

comparison is using a spider (also known as radar) chart, shown in Fig. 2.7. This way of comparing the implementation quality lets the developer see each aspect of the implementation quality in detail.

The order of quality components does not affect the total quality, therefore they can be sorted from highest to lowest in the spider chart to improve readability. When there are multiple qualities shown in the chart, sorting can be done according to the quality that is being highlighted. That way it is easier to see which quality components differ the most.

In the case where the total implementation quality is important instead of the separate qualities Q, a single numeric expression of Q_{MP} is needed. To compare implementation qualities of different MP systems or different implementation methods, the average quality can be used:

$$\overline{Q}_{\mathrm{MP}} = \sqrt[5]{Q_{\mathrm{A}}^{\mathrm{MP}} \cdot Q_{\mathrm{T}}^{\mathrm{MP}} \cdot Q_{\mathrm{R}}^{\mathrm{MP}} \cdot Q_{\mathrm{P}}^{\mathrm{MP}} \cdot Q_{\mathrm{I}}^{\mathrm{MP}}}\,. \tag{2.21}$$

The use of the fifth root or the product of Q values allows to better represent the cases where one of the components significantly differs from the others, compared to just calculating the average value. To take Fig. 2.7 as an example, $\overline{Q}_{\mathrm{MP1}}$ would be:

$$\overline{Q}_{\mathrm{MP1}} = \sqrt[5]{10 \cdot 50 \cdot 90 \cdot 5 \cdot 22} = 21.8\,. \tag{2.22}$$

Although the average quality can be used to compare different implementations of MP systems, there are cases where another parameter is needed. Two different compared quality $\overline{Q}_{\text{MP}}$ values can be similar, although their components Q can differ significantly. For example, let's consider Q_{MP2} and Q_{MP3} in that are shown in Fig. 2.7:

$$\overline{Q}_{\text{MP2}} = \sqrt[5]{80 \cdot 20 \cdot 55 \cdot 20 \cdot 35} = 36.1\,; \tag{2.23}$$

$$\overline{Q}_{\text{MP3}} = \sqrt[5]{30 \cdot 40 \cdot 33 \cdot 39 \cdot 44} = 36.9\,. \tag{2.24}$$

For this reason when average quality of implementations is similar as is this case, the dispersion of Q values should be considered:

$$\sigma_{\text{MP}} = \sqrt{\frac{1}{|Q_{\text{MP}}|} \sum_{Q \in Q_{\text{MP}}} (Q - \overline{Q}_{\text{MP}})^2}\,. \tag{2.25}$$

When comparing different implementations, if the dispersion in (2.25) is lower, it means that the quality is more even across the different Q values. In most cases that is the desired result. This way it can be determined that implementation quality of MP3 system is better, as its dispersion is significantly lower:

$$\sigma_{\text{MP2}} = \sqrt{\frac{1}{5} \sum_{Q \in Q_{\text{MP2}}} (Q - 36.1)^2} = 24\,; \tag{2.26}$$

$$\sigma_{\text{MP3}} = \sqrt{\frac{1}{5} \sum_{Q \in Q_{\text{MP3}}} (Q - 36.9)^2} = 5\,. \tag{2.27}$$

The direct comparison of MP system implementations in FPGA, as demonstrated in Fig. 2.7, is only possible when both implementations use exactly the same FPGA model. This is because the calculation of quality uses these parameters of FPGA as boundaries: $f^{\top}$, N_{maxLUT}, N_{maxIOB}, P_{TDP}. That means that in case of implementation in different chips all aspects of implementation quality, except for accuracy Q_{A}, must be transformed before comparison.

To transform the calculated implementation quality Q to the frame of reference of another FPGA model, a special coefficient should be used:

$$Q^* = k_\theta \times Q\,, \quad Q \in Q_{\text{MP}}\,, \tag{2.28}$$

with the coefficient of transformation equal to

$$k_\theta = 1 - \frac{\theta_1 - \theta_2}{\max(\theta_1, \theta_2)}\,, \tag{2.29}$$

where Q^* – quality of one aspect of implementation transformed into the reference frame of another FPGA model; θ_1 – value of boundary parameter of current reference

frame; θ_2 – value of boundary parameter of the reference frame that the quality in being transformed into.

Example 2.1 (MP system quality transformation)

Let's transform $Q_{\mathrm{R}} = 50$ *from Xilinx Artix-7 XC7A50T FPGA* ($N_{\mathrm{maxLUT}} = 75680$) *reference frame to Xilinx Artix-7 XC7A75T FPGA* ($N_{\mathrm{maxLUT}} = 110800$) *reference frame:*

$$Q_{\mathrm{R}}^{*} = k_{\theta} \times Q_{\mathrm{R}} = (1 - \frac{75680 - 110800}{110800}) \times 50 \approx 66\%, \tag{2.30}$$

here as expected Q_{R}^{*} *is higher than* Q_{R} *because the amount of used resources is lower as a percentage of total FPGA resources. To avoid cases where quality exceeds* 100 %, *it is recommended to transform quality from smaller FPGA reference frame to larger FPGA reference frame.*

2.6. Conclusions of the Second Chapter

After defining the quality evaluation criteria of Metabolic P system implementation in hardware, the following conclusions can be made:

1. Implementation quality of Metabolic P systems in FPGA can be adequately estimated by comparing accuracy, throughput, resource usage, power consumption and interface complexity.
2. Hardware implementations use fixed point calculations that suffer from decreased accuracy compared to floating point calculations. Therefore to evaluate accuracy a reference implementation in software is needed.
3. The binary word length used for fixed point arithmetic determines the RMSE value of the calculations. The required word length can be found when the desired accuracy is known by using a proposed algorithm.
4. Hardware resource usage is always a compromise between speed, spatial resources and power consumption, as demonstrated by Fig. 2.7. The most desirable implementation should always be chosen by the developer based on the limitations of used hardware.
5. Proposed average quality and dispersion can be used when comparing different implementations in the same FPGA. Quality values must be transformed to another reference point when comparing implementations in different FPGA models.

3

Metabolic P Systems Hardware Implementation Techniques

In the following chapter the theoretical results of MP systems as well as best FPGA implementation practices are used to offer original solutions for MP system transformation to FPGA structural elements. First, the available techniques for Metabolic P system implementation in FPGA are analyzed. Three common techniques for Metabolic P system implementation are presented in Section 3.1, including combinative (Section 3.1.2), single DSP cell (Section 3.1.3) and pipelined (Section 3.1.4) implementations. A new developed data structure that uses JSON format is presented in Section 3.1.1. A novel technique for unification of multiple MP systems is proposed to improve the quality of implementation of several similar MP systems at the same time. The unified pipelined implementation technique is presented in Section 3.2.1. The VHDL code examples of the novel implementation technique are presented. The developed technique is then demonstrated by implementing a group of previously described IVGTT MP systems (Section 3.2.2). The structure of the unified pipelined IVGTT implementation that demonstrates the number of calculation stages is presented.

The research results are published in author publications (Kulakovskis, Navakauskas 2015, 2016; Kulakovskis *et al.* 2016; Kulakovskis 2019). The main results are announced in international AIEEE (Riga, 2015) and national "Science – Future of Lithuania" (Vilnius, 2016, 2017) scientific conferences.

3.1. Common Techniques for Metabolic P System Implementation

The possibilities and challenges of implementing Metabolic P systems in hardware are examined by Guiraldelli, Manca (2015a). There it is proven that MP systems can be directly translated to register machines and the implementation in hardware is suggested as the next step.

There are multiple different methods that can be used for MP system implementation in FPGA. In Section 1.4 MP system implementation in software ways were described and the possibilities of implementation in hardware were discussed. It was shown that there are no known implementations of MP systems in FPGA. Therefore let's start by implementing MP systems using various common techniques. Those techniques have advantages and disadvantages that can be useful for a particular application. The specifics of MP systems also play a part in the implementation methods. They must be taken into account when determining the suitability of implementation technique.

3.1.1. Metabolic P System Representation by Data Structure

To better describe MP systems in a format that is easily interpreted by programming logic, a JSON data structure of original design was developed. The JSON format was chosen because it is easy to read and convert in most modern programming languages. Also, it is a relatively simple format used by many applications, especially web pages that use JavaScript language. The structure of the JSON file closely mirrors the structure of MP systems represented by MP graphs. This feature makes it easier to convert MP systems from one format to another by automated tools or by manual human interaction.

The JSON data structure contains all required information to describe an MP system. An imaginary MP system described by the MP graph shown in (2.10) can be represented in the JSON format of Example 3.1.

Example 3.1 (MP system represented by JSON data structure)

mp.json

```
{
  "substances": {
    "A": 1,
    "B": 2
  },
  "rules": [{
    "add": {
      "B": 3
    },
    "sub": {
      "A": 2
```

```
    }
  }, {
    "add": {
      "A": 1
    },
    "sub": {
      "B": 1
    }
  }],
  "constants": {
    "c1": 5,
    "c2": 8.5,
    "c3": 1
  },
  "fluxes": [
    ["c1", "*", "A", "+", "c2"],
    ["A", "+", "B", "-", "c3"]
  ]
}
```

This example MP system contains two of each substances, rules and fluxes. In MP grammar the count of these elements is always equal. Only the described substances can be used in rules object. Each rule has a `sub` *and* `add` *parts that determine the amounts of substances that are added or subtracted from the system by each rule. There are also three constants in this example system. There can be any number of constants defined, but only the defined constants can be used in the* `fluxes` *object together with the defined substances.*

When implementing the MP system in a new JSON format, in the code it is split into four parts: substances, rules, constants and regulators. Let's analyze them more thoroughly.

Substances are the main variables in MP systems. In this case there are two substances: `A` and `B`. These substances must have an initial amount that will be used at the start of MP system calculation. In JSON data structure keys of substances object represent the substances of MP system and the objects values represent the initial amount of corresponding substance.

Rules are the main MP system reactions that transform one or more substances into each other, introduce substances from the environment or expel them to the environment. In this case there are two reactions. The first one transforms an amount of two of substance `A` into an amount of three of substance `B`. In other words, `2A` is subtracted and `3B` is added. This is represented in JSON data structure by a one dimensional array that has a length of the number of rules in the MP system. It contains the two objects (each for one rule) with numbers of added and subtracted amount of each substance as key-value pairs.

Constants are the numerical values that do not change during the MP system iterative calculation process. Constants participate in the addition, subtraction or multiplication operations of MP system regulators. In JSON data structure constants are

represented in a constants object by key-value pairs where the key is an arbitrary constant name and value is the constant itself. This enables the separation of values from mathematical operations. In this case there are three constants represented by two integer and one fractional numbers. These are the only allowed types of constants as the MP system must be implemented in hardware that has its limitations.

Regulators, sometimes also called fluxes, determine the rate at which the reactions are happening. Each MP system rule has its own regulator expression that contains mathematical operations using the previously defined constants and substances. In the JSON data structure allowed mathematical operations are addition, subtraction and multiplication. More complex operations could in principle be implemented, but hardware limitations must always be considered. In JSON data structure each regulator is represented by an array that contains each operand and mathematical operation of the regulator expression as a separate array element. The array must start and end with a substance or constant (valid mathematical operations cannot start or end with a sign) and elements must be separated by addition (+), subtraction (−) or multiplication (*) signs.

3.1.2. Combinative Implementation

The combinative implementation, also known as combinational logic implementation, uses traditional behavioral process with synchronous reset. Combinative implementation uses FPGA elements like slices, LUTs and DSP cells. The exact configuration of FPGA elements is decided when optimization and synthesis are performed by the VHDL code synthesizer.

A. Operation Sequence in Combinative Implementations

VHDL code of single behavioral process with synchronous reset is used to implement MP system using combinative technique. The use of FPGA slices, LUT and DSP cells is optimized and by the code synthesis tool that is provided by the FPGA chip manufacturer.

The key point of combinative implementation technique is that all MP calculations are performed on the same clock cycle. This means that to achieve this, all the logic elements of FPGA that perform calculations need to be connected in a long chain. The intermediate values in the calculation are used as an input values of the next calculation step. After a single clock cycle the next MP system value is available.

The VHDL code of combinative implementation uses a single sequential process to describe the data processing routine as shown in Example 3.2. The MP calculations are all connected sequentially according to implemented MP system specifics. VHDL variables are used to store temporary results that can be used again at the same clock cycle. Finally, when the next output value of MP system is calculated, the result is assigned to a signal and the step counter is incremented.

Example 3.2 (Combinative implementation of IVGTT MP system)

combinative.vhd

```
if rising_edge(clock) then
  rule_1 := f1c1;
  rule_2 := (f2c1 * G) + (f2c2 * G * G * I);
  rule_3 := f3c1 + (f3c2 * G * G * G);
  rule_4 := f4c1 * I;

  I_var := I + rule_3 - rule_4;
  G_var := G + rule_1 - rule_2;

  I <= I_var;
  G <= G_var;
end if;
```

The combinative implementation consists of three main steps. First, each of the four rules of IVGTT MP system are calculated and stored in variables. Then substance variables `I_var` *and* `G_var` *for insulin and glucose are calculated. And finally output signals* `I` *and* `G` *are assigned the calculated values. All this sequence happens during one clock cycle.*

The main advantage of combinative implementation technique is that the latency of calculation always equals one clock cycle. This means that calculations are performed sequentially in the same clock cycle. However, this also means that the maximum frequency of the implementation can decrease significantly. This effect can be especially seen when a more complex MP system is implemented because more calculations need to be performed during the same single clock cycle. This must be taken into account when investigating implementation results.

B. Technique for Automated Combinative Implementation

The developed automated implementation technique was implemented in MP system VHDL code generation tool written in Python interpreted programming language (Kulakovskis, Navakauskas 2016). It was developed to convert MP systems described by a data structure in JSON format to VHDL architecture. The generated VHDL code can be implemented as a stand-alone component or incorporated into bigger programs that are implemented in FPGA. This enables easier implementation of MP systems in hardware and builds on key MP system aspect of accessibility and ease of description. The tool can be used not only by engineers who understand MP systems, but also people working in different fields who might find functionality provided by MP systems useful but too time and resource consuming to develop independently. The developed tool allows to achieve faster implementation and lowers the time needed to deploy a selected MP system or even a multitude or variation of them.

The MP system VHDL code generation tool is composed of four main parts, as shown in Fig. 3.1 on the next page.

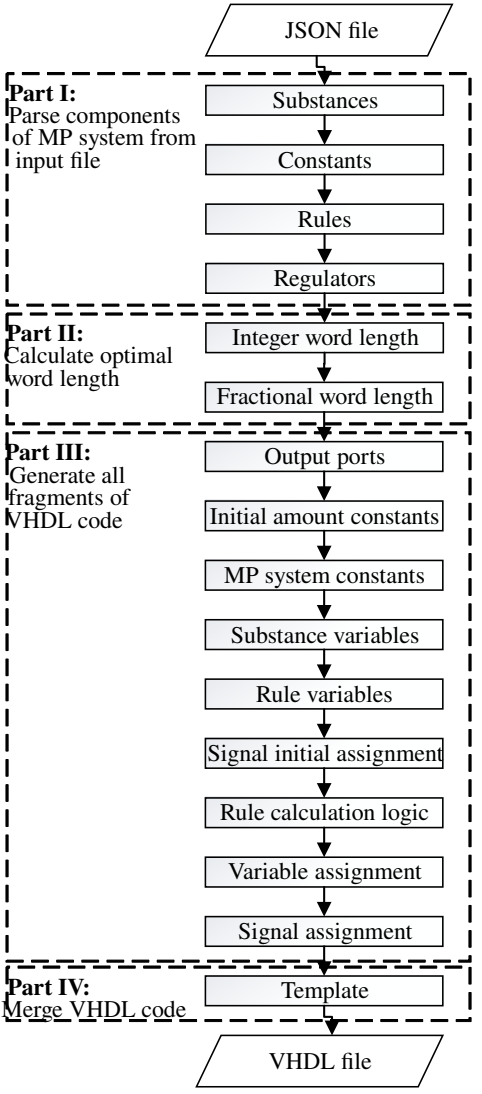


Fig. 3.1. The structure of metabolic P system code generation tool

The first part of the tool consists of an input file interpreter and validator. The second part uses the input data and determines the binary word length needed to achieve the requested accuracy. The third part of the tool uses the determined word length and the input data to generate key parts of VHDL code. Lastly, the fourth and final part merges the generated VHDL code fragments with a prepared template and produces a complete VHDL component.

As an input a JSON data structure, described in Section 3.1.1, is used to describe MP systems. The file includes entries for initial substance amounts, constants, rule logic and regulator logic. Also, the developed tool accepts an input of a number of iterations the software should generate and the desired accuracy of the resulting fixed point calculations. The accuracy is used to determine the word length used in VHDL fixed point library.

The second part of the MP system VHDL code generation tool determines the optimal word length that should be used to achieve the calculation accuracy requested by user. The word length is determined by an algorithm described in Section 2.5.1.

The third part of the MP system VHDL code generation tool is the main code generation logic. It uses the input MP system data as well as determined integer and fractional word lengths to generate specific sections of VHDL code. The generation part consists of nine steps, as shown in Fig. 3.1.

Firstly, an output port must be defined for each substance participating in the provided MP system. The port direction is set to bidirectional (`inout`) and signed fixed point (`sfixed`) type is used. Then the constants used in this VHDL component are defined. They consist of substance initial amounts, taken from the substances part of JSON file, and other constants used in MP system regulator expressions, defined separately in JSON file. Some variables must be defined for the behavioral process that calculates a step of MP system values on each clock cycle. Substance variables store the state of substance amounts after each calculation step (clock cycle). Rule variables store the calculation result for each MP system rule. When the VHDL process begins, firstly MP system substance signals are initialized by passing them the previously defined initial substance amount constants. Then the main calculation logic starts. For each MP system rule a single line expression is generated that implements the regulator logic. Then from the calculated rules substance amount in for the current step are determined and assigned to substance variables. Finally, the calculated substances are passed to signals that can be sent to the output.

The fourth and last part of MP system VHDL code generation tool uses a prepared template containing the rest of VHDL code such as library declarations, utility port (step number, clock, start signal, etc.) declarations, rising clock edge detection and start and end markers of various VHDL code blocks. These code fragments are static for every implemented MP system and do not have to be generated. In addition the template contains replacement fields that are substituted with generated VHDL code fragments by using Python string formatting tools. The result of merging the template with generated code fragments is a complete VHDL component of a particular MP system provided as an input.

3.1.3. Single Digital Signal Processor Slice Implementation

All recent FPGA have dedicated DSP slices for fast arithmetic implementation. Such a DSP block can be shared to process data from different sources and can be dynamically reconfigured to implement different processing orders (Xilinx 2014).

The DSP slice accepts the configuration according to:

$$P = C|P \pm (D \pm A) \times B\,, \tag{3.1}$$

where P – the output value of DSP slice; A, B, C, D are the input values of DSP slice; $|$ – alternate operator.

The DSP block is purposely designed for pipelined data processing with maximum performance. This is achieved by inserting flip-flops and minimizing critical path in a circuit (Fig. 3.2). Sequentially connected flip-flops introduce latency of 4 clock cycles in the data flow processing independently of the selected instantaneous DSP operation code and subgraph constellation Xilinx (2014).

A. Metabolic P System Representation by Data Flow Graph

MP systems contain addition, subtraction and multiplication operations. All these operations are supported by the DSP. Therefore, MP systems can be represented as a data flow graph, where the graph nodes express arithmetic operation. The aim is to partition whole graph into DSP slice suitable subgraphs and then cover the graph with largest detected subgraph. The DSP slice subgraph has maximum four input terminal nodes and a single output terminal node. The source node can be connected with a destination node to create a subgraph only if the source node has no branches to other nodes except the branch to destination node. Such a constraint follows from physical structure of the DSP slice. It is reasonable to find largest subgraphs first, as such approach decreases total number of subgraphs resulting in minimal latency and reduces resource usage, since the internal signals in DSP do not need to be stored in registers outside DSP (Sledevič, Navakauskas 2015).

The MP system implementation on FPGA technique is shown in Fig. 3.3. It is based on the conversion of the equations to the DFG and splitting it to the DSP supportable subgraphs and further scheduling of DSP operations.

An example of the IVGTT (described in Section 1.3.3) DFG and its partition-

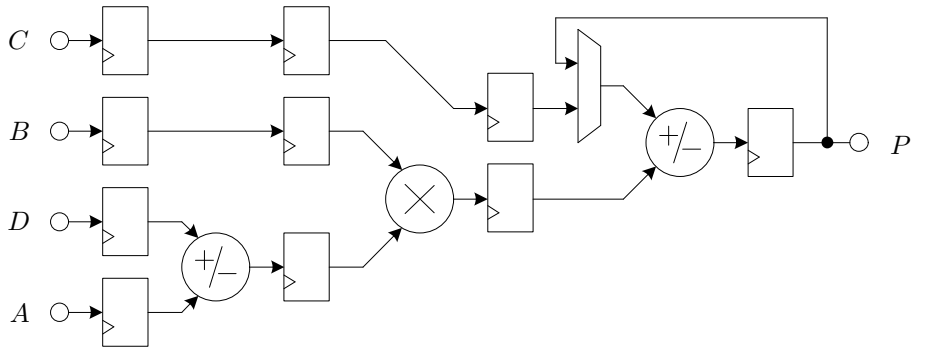


Fig. 3.2. The simplified structure of digital signal processor slice

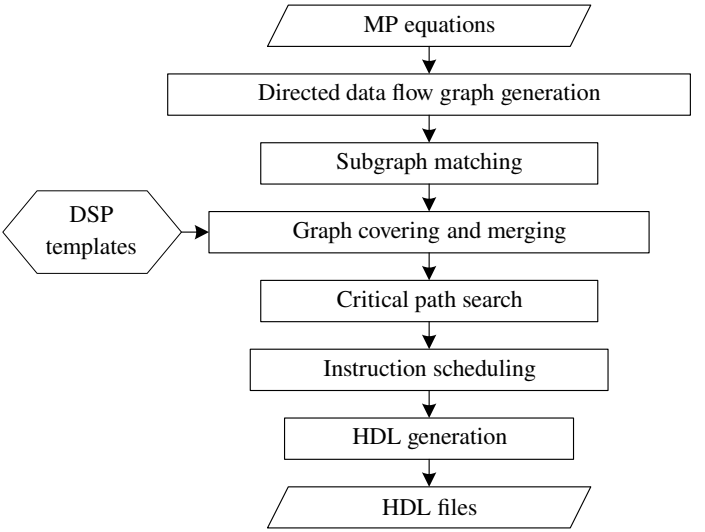


Fig. 3.3. The technique for metabolic P system implementation in field programmable gate array based on single digital signal processor slice

ing (marked by grey background) is shown in Fig. 3.4. At the DFG generation stage the operators are converted to a set of vertices connected by set of edges in a directed graph. The DFG is stored as an adjacency list. At the subgraph matching stage all the DSP supportable subgraphs described by (3.1) must be found in a graph. The subgraph is detected in a graph if there exist one-to-one correspondence between vertices and edges. The depth-first search strategy with respect to DSP constraints is implemented for subgraph matching. Therefore, the subgraph is registered and indexed if it has only single branch between any two internal nodes.

The graph covering and merging stage is a DSP supportable subgraphs selection process, that attempts to find an appropriate set of subgraphs which minimizes resources usage and latency. The largest subgraphs detected in graph are covered first using greedy algorithm (Cong, Jiang 2008; Ronak, Fahmy 2014). It checks iteratively a presence of largest subgraphs first in an MP system graph scanning it through all matched subgraphs. The covered subgraphs are registered. Afterwards, all subgraphs with overlapping vertices are eliminated from a set of candidates for further covering.

The DSP has a property to add/subtract output value P of previous clock cycle through the multiplexed output feedback to last vertex of DSP subgraph, as shown in Fig. 3.2. Therefore, DSP must be reconfigured to access output P instead of input C (3.1). Such DSP configuration allows to process a subgraph in a single clock cycle due to the pipelined subgraph processing. In order to apply the pipelining for two or more adjacent subgraphs, the pairs of subgraphs must be merged, when a common edge exists between two terminal vertices. As an example, two merged sets of subgraphs inherent for IVGTT graph are distinguished. Subgraphs within the sets

$\{7, 2, 4\}$ and $\{1, 9, 3\}$ are merged. The DSP instructions for subgraphs with indices $2, 4, 9, 3$ are modified to $P = P + D \times B$, $P = P - D + A$, $P = P + D$, $P = P - D \times B$ respectively. All modified instructions will be executed in single cycle. The covered and merged subgraphs are returned for further scheduling. The merged subgraphs are connected by grey curves in Fig. 3.4.

B. Field Programmable Gate Array Resource Scheduling

The arithmetic resources of FPGA must be shared through all subgraphs simultaneously exploiting the pipeline property of DSP. Moreover, each instruction must be exactly placed in time maintaining shortest latency for whole covered graph execution. It is known, that for resources unconstrained scheduling the critical path method gives a shortest possible delay limited only by data dependence in critical path. For the reason that the result of 5th subgraph is needed by 1st and 6th subgraphs before it has finished executing, the pipeline hazard occurs (the 1st and 6th subgraphs must wait for the 5th to complete).

Giving a set of covered graphs the critical path method returns: the minimal latency needed for subgraphs computation, the set of subgraphs located on critical path and the vector with early start time values. The subgraphs, that are on the critical

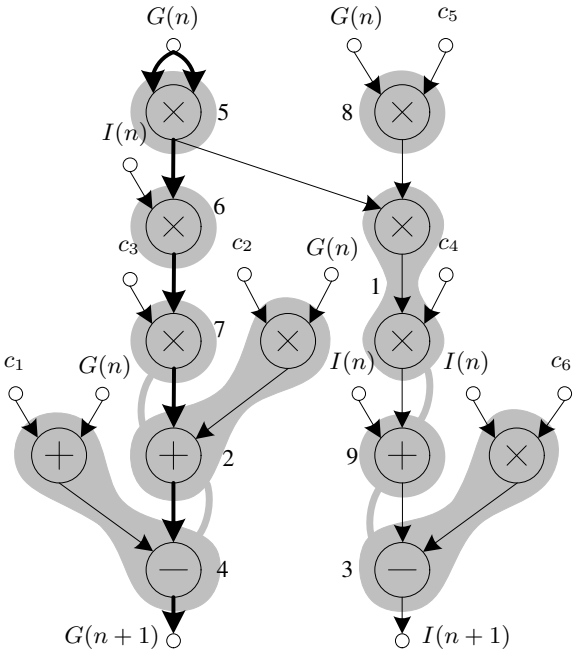


Fig. 3.4. A graph of the metabolic P system model and its partitioning into subgraphs (in grey). The merged subgraphs are connected by curves (in grey)

path (bold arrows – Fig. 3.4, connected instruction – Fig. 3.5), must be executed with highest priority, since the latency is directly related with delay in critical path. The priority of all other subgraphs, that lie not on critical path, depends on the early start time. The earlier the subgraph starts, the higher is its priority.

The Gantt chart for resources constrained scheduling is shown in Fig. 3.5. The latency of IVGTT graph calculation is shortest possible. It can be seen from the sequence of the scheduled instructions located exactly one after another in critical path and connected by directed arrows. The dashed rectangles mark slack for instructions, that not influence the total latency. The subgraph scheduling algorithm handles pipelined DSP resources by allowing the scheduling of overlapping operations with no data dependency and different start times. If two or more operations have same start time, then they are delayed until DSP will be released. The developed scheduling algorithm includes the candidate subgraph to set of subgraphs on the schedule list only if all predecessor vertices for candidate subgraph are computed. Then it schedules exactly one subgraph from a set of candidates with highest priority. Thus the subgraphs located on critical path are scheduled first before all other subgraphs. The subgraphs $\{8, 1, 9, 3\}$, that not belong to critical path have a slack of free time steps and are scheduled with as soon as possible strategy. The proposed scheduling algorithm returns instruction start vector of time steps, that is used next in HDL generation placing DSP instructions in right order.

The final architecture of processing element described in HDL is shown in Fig. 3.6. It contains single DSP as a configurable subgraph execution unit and four buffers connected to DSP inputs. The index generator consists of binary counter. With the instruction counter a proper instruction is selected at each clock cycle from a set of instructions. The instruction set format consists of: four write/read addresses, DSP

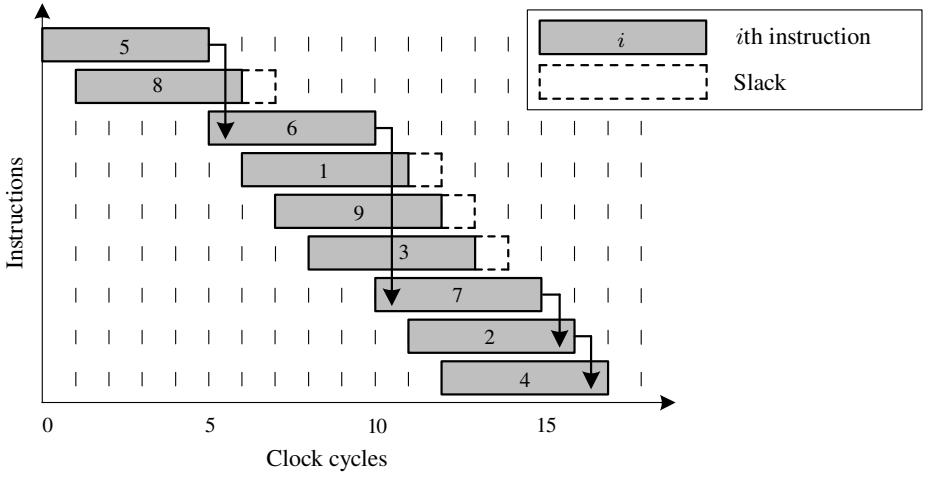


Fig. 3.5. The resource constrained schedule for metabolic P system graph

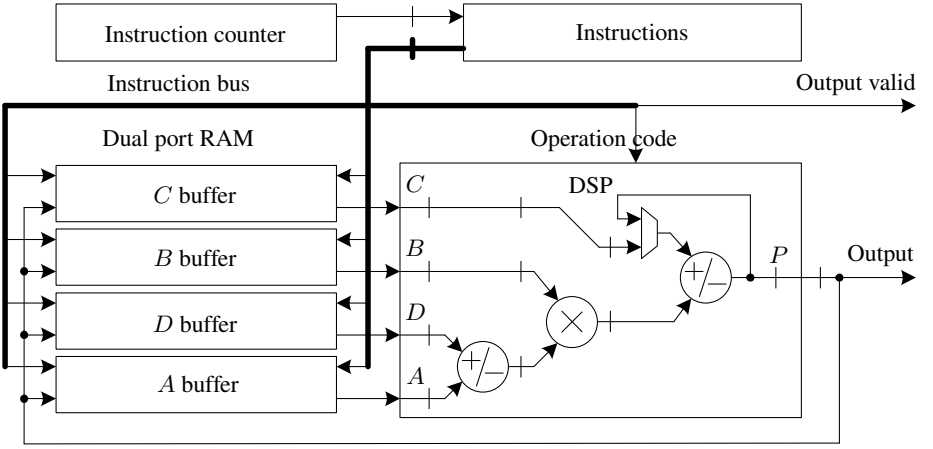


Fig. 3.6. The processing element based on single digital signal processor slice

operation code bits and output valid bit. The data buffers are implemented on distributed dual port memory. The DSP output P is returned back to buffers and is connected to output.

Due to the simplification the delay elements in Fig. 3.6 are marked by "|" crossing data lines. The four cycle latency of DSP are determined by hardware constraints. The timing analysis after place and route stage gives the maximum clock frequency $f_{\max}$ at which the implemented design can run. The time optimal instruction execution was achieved with one additional flip-flop (delay element) located on the DSP output.

3.1.4. Pipelined Implementation

The pipelined MP system implementation takes advantage of the way the MP system calculations are structured. Often there are multiple calculations that can be executed in parallel. This allows to use the parallel computation advantage of hardware implementations.

In case of pipelined implementations the computation of the next MP system values proceeds in a pipeline. Due to the data dependence in the MP system physical structure, the computation of new value can start only when the previous value is computed. However, the pipelined structure allows to utilize signal interleaving feature and compute values simultaneously for multiple channels.

The main difference of the pipelined implementation compared to the combinative implementation is the use of signals to store intermediate values instead of variables. This allows one MP system calculation step to be performed in multiple clock cycles instead of a single long cycle. Therefore the maximum frequency of the design can be increased.

The exact structure of the pipelined implementation depends on the implemented MP system. The number of stages (clock cycles) is determined by the complexity of fluxes and the amount of substances. Also, only certain calculations can be performed in parallel as the intermediate result if often needed for the next operation.

The example as soon as possible operation schedule is used for pipelined IVGTT implementation, as shown in Fig. 3.7.

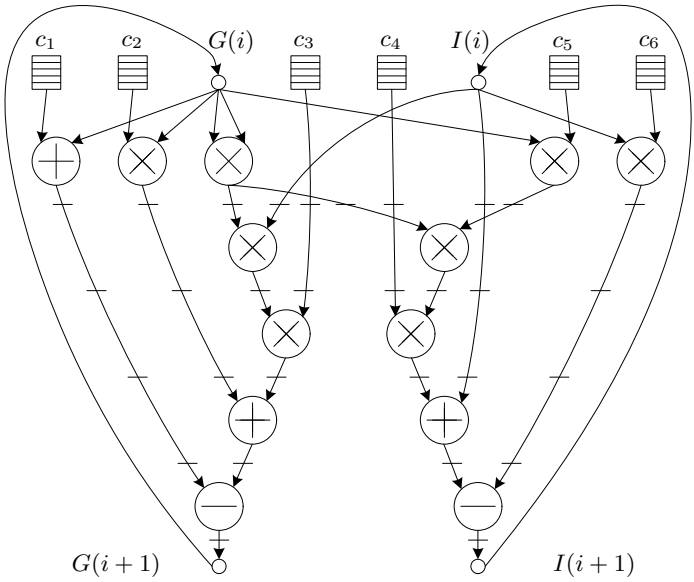


Fig. 3.7. Illustration of pipelined implementation of intra-venous glucose tolerance test metabolic P system for interleaved data processing

The $G(n+1)$ and $I(n+1)$ are valid after five clock cycles. Due to the pipelined structure the output values can be calculated for five different channels employing multi-channel data interleaving. For this purpose shift registers are dedicated for each coefficient c_i, $i \in \{1, \ldots, 6\}$.

3.2. Novel Technique for Unified Metabolic P System Implementation

The common techniques discussed in Section 3.1 are well suited when implementing a single MP system in FPGA (Kulakovskis *et al.* 2016). However, often using FPGA for calculating one system at a time is inefficient. There are use cases where many MP systems need to be calculated simultaneously, either for increased reliability or calculation of multiple research objects. In this case the resource usage of common

techniques scales linearly with the number MP systems that need to be calculated. The technique of unification of multiple implementations can be used to increase the efficiency of the implementation of many variations of any MP system type.

3.2.1. Unified Implementation Technique

The main goal of the unified MP system implementation is to build a general MP system that covers the different variations of a particular MP system type. These variations include different coefficients or even different arithmetic operations in fluxes. The static parameters are the number of rules and substances. Both the pipelined and combinative techniques can be used for unified MP system implementation.

Example 3.3 (Selection of a set of MP system parameters)

mp_library.vhd (fragment)

```
13 package mp_library is
14   constant wi : integer := 64; -- total signal width
15   constant le : integer := 31; -- sig. width start
16   constant ri : integer := wi-le-1; -- sig. width stop
17   constant Ginit : integer := 19; -- initial G value
18   constant Iinit : integer := 180; -- initial I value
19
20   type coef_type is
21     array (1 to 6) of sfixed(le downto -ri);
22   constant c1_mem : coef_type := ( -- coefficient c1
23     to_sfixed(0.12, le, -ri),
24     to_sfixed(0.12, le, -ri),
25     to_sfixed(0.0, le, -ri),
26     to_sfixed(0.00052, le, -ri),
27     to_sfixed(0.02, le, -ri),
28     to_sfixed(0.0, le, -ri));
```

Omitted constants `c2–c10` *follow the same pattern – each constant is an array of six* `sfixed` *numbers.*

mp_library.vhd (fragment)

```
90   type mux_type is
91     array (1 to 6) of std_logic_vector(1 to 8);
92   constant mux_mem : mux_type:=( -- multiplexer
93     "1XXX11X0", "10XXX000", "0XXXXX10",
94     "01111111", "00011110", "01100100");
95 end mp_library;
```

The array of multiplexer values is the same length as the array of coefficients. This way the appropriate coefficients and multiplexers can be selected by using an index value that corresponds to the exact MP system. The length of the arrays is equal to the number of MP systems defined in the library. This means that theoretically any number of systems can be defined in the library (as long as these is space left in FPGA) and later selected by input signal.

The different variations of the same type of MP system usually have similar input parameters and the order of arithmetic operations. Therefore, the branches in a pipelined implementation with same computational resources can be enabled, disabled or shared with several MP systems. This way the application of unified MP system is advantageous when multiple different MP systems need to be implemented on a single FPGA. The specific MP system to be calculated can be selected using input parameter in hardware description model.

Example 3.4 (Main process of unified pipelined implementation)

mp.vhd (fragment)

```
80 process(clk)
81 begin
82   if rising_edge(clk) then
83     case state is
84       when st_0 => -- initial stage
85         if start = '1' then
86           state <= st_1;
87           G_sum <= to_sfixed(G, le, -ri);
88           I_sum <= to_sfixed(I, le, -ri);
89           G_neu <= to_sfixed(G, le, -ri);
90           I_neu <= to_sfixed(I, le, -ri);
91           done_fsm <= '1'; -- indicate results
92         end if;
93         enable_shift_reg <= '0';
94       when st_1 => -- stage 1
95         state <= st_2;
96         if enable_shift_reg = '1' then
97           -- update shift registers
98           for i in w64 downto 3 loop
99             Gshreg(i) <= Gshreg(i-1);
100          end loop;
101          Gshreg(2) <= G_sum;
102        end if;
103        done_fsm <= '0';
```

The MP calculations of the first stage are omitted to save spacepe. The calculations are performed up to stage 10*. On each stage all operations that are independent can be performed.*

mp.vhd (fragment)

```
259      when st_10 =>
260        state <= st_1;
261        G_neu <= G_new;
262        G_sum <= resize(G_sum + G_new, p_fix);
263        I_neu <= I_new;
264        I_sum <= resize(I_sum + I_new, p_fix);
265        done_fsm <= '1';
266        enable_shift_reg <= '1';
267      when others =>
268        state <= st_0;
269    end case;
```

```
270   end if;
271 end process;
```

In this example the whole process only consists of 10 *calculation stages. That means the results of the MP system calculation step are ready after* 10 *clock cycles. The external availability of the results is indicated by* `done_fsm` *signal. It is set to* `1` *only on the last stage at* ***L265*** *(and initial, as a special case in* ***L91****) stage. In the* ***L103*** *the signal is again set to* `0`.

The settings for all MP systems used in calculations are stored in a common VHDL library as shown in Example 3.3. Each coefficient of MP system is defined as an array of constants. The length of the array is equal to the number of different variations of the MP system.

An important aspect of the implementation is that all coefficients must be defined for all systems, even if a particular system uses a more simple regulator function that has fewer operations. In such case the coefficient is set to zero and the operation is not performed. The coefficients are stored in `sfixed` format that indicates that signed fixed point arithmetic is used.

The values for multiplexers are stored along with the coefficients. The implementation technique uses multiplexers for selection of the correct calculation path for each MP system. The values of `1` and `0` determine the state of the multiplexer and the used path. The value `X` is used to signify that the multiplexer is not used in this particular system. The standard logic vector of multiplexer states is separate for each MP system and is selected in the same way as the coefficients. The length of the logic vector is equal to the total number of multiplexers in the implementation.

The main process of the pipelined implementation consists of a number of stages. On each stage a maximum number of independent calculations is performed. The number of stages matches the number of clock cycles needed to get the next values of MP system substances. This way the highest sample generation frequency is achieved. There is also an extra initial stage that is executed only once at the start of calculations of an MP system or when switching to another system, as shown in Example 3.4. This process is very similar to the one described in Section 3.1.4.

One of the main differences of the unified pipelined implementation is that an extra behavioral process is introduced. As shown in Example 3.5, it uses an external selector signal to select the proper coefficient and multiplexer values as defined in Example 3.3. This allows the implementation to rapidly switch from calculating one MP system to another, without needing to implement separate logic for each system.

Example 3.5 (Additional process for coefficient selection)

mp.vhd (fragment)

```
62 process(clk)
63 begin
```

```
64   if rising_edge(clk) then
65     L_sel <= sel; -- external selector
66     mux <= mux_mem(to_integer(unsigned(L_sel)));
67     c1 <= c1_mem(to_integer(unsigned(L_sel)));
68     c2 <= c2_mem(to_integer(unsigned(L_sel)));
```

The omitted coefficients `c3 – c10` *are set in the same way as shown above. The* `L_sel` *signal indicates the selected MP system index according to the systems defined in the library.*

The unified combinative implementation has a very similar structure to the pipelined implementation. The key difference is that, as with combinative single MP system implementation technique, all calculations are executed during one clock cycle. This means that the MP system type can be switched after a single substance value is calculated. The potential disadvantage of unified combinative implementation is that all types of MP systems must be able to be calculated by a single long logic chain. This may result in decreased maximum system frequency.

The implementation can also optionally include shift registers as shown in Example 3.4. They are needed in case the implemented MP system uses substance values that are older than one clock cycle in its calculations. The shift registers provide this function and are updated on the first stage of the process.

The shared resources of the implementation mean that different MP systems can be calculated either one after another or even simultaneously. To achieve that the different MP systems can be mixed during the interleaved computation. In that case the coefficients and the select values for multiplexers must be updated on each clock cycle.

3.2.2. Unified Intra-venous Glucose Tolerance Test Metabolic P System Implementation

The IVGTT system has multiple variants of MP grammar. Metabolic P systems are defined by approximating different sets of patient data. Six different MP systems that were defined by Manca *et al.* (2011) are used for FPGA implementation research. Although MP grammar of different IVGTT MP systems is not identical, the structure is very similar and the differences are primarily in constant coefficient values. This would also be true for any new IVGTT MP systems derived from new experimental patient data. The common equation for calculating glucose and insulin values for any IVGTT MP system uses four fluxes, one of which is a constant:

$$\begin{aligned} G_n &= \phi_1 - \phi_2(G_{n-1}, I_{n-1}); \\ I_n &= \phi_3(G_0, \ldots, G_{n-1}) - \phi_4(I_{n-1}), \end{aligned} \tag{3.2}$$

where G_n – glucose; I_n – insulin; ϕ_1, ϕ_2, ϕ_3, ϕ_4 – fluxes, shown in Table 3.1.

Table 3.1. Fluxes of used intra-venous glucose tolerance test metabolic P systems

MP system	Fluxes
IVGTT-1	$\phi_1 = 0.6$ $\phi_2 = 0.12G + 1.6 \cdot 10^{-6} G^2 I$ $\phi_3 = 49.9 + 0.1G^3$ $\phi_4 = 0.84I$
IVGTT-2	$\phi_1 = 0.6$ $\phi_2 = 0.12G + 1.6 \cdot 10^{-6} G^2 I$ $\phi_3 = 1.5 \cdot 10^{-5} G^6 + 0.25G^2_{-6} + 0.17G^2_{-8} + 2.65G_{-16} +$ $+ 3.6G_{-26}$ $\phi_4 = 0.65I$
IVGTT-3	$\phi_1 = 0.011$ $\phi_2 = 6.6 \cdot 10^{-5} GI$ $\phi_3 = 0.5G^2_{-4}$ $\phi_4 = 0.16I$
IVGTT-4	$\phi_1 = 0.056$ $\phi_2 = 5.2 \cdot 10^{-4} I + 8.1 \cdot 10^{-5} GI$ $\phi_3 = 3.76 \cdot 10^{-6} G^7 + 0.74G^2_{-8} + 0.02G^3_{-20} + 0.21G^2_{-40} +$ $+ 10^{-4} G^5_{-68}$ $\phi_4 = 0.49I$
IVGTT-5	$\phi_1 = 0.12$ $\phi_2 = 0.02G + 1.9 \cdot 10^{-4} GI$ $\phi_3 = 0.04G^3_{-2} + 3.3 \cdot 10^{-5} G^6_{-6} + 0.44G^2_{-20} + 0.04G^3_{-24}$ $\phi_4 = 0.5I$
IVGTT-6	$\phi_1 = 0.11$ $\phi_2 = 6.2 \cdot 10^{-4} GI$ $\phi_3 = 0.1G^2_{-2} + 0.9G_{-6} + 1.07G_{-10} + 2.4 \cdot 10^{-4} G^4_{-24} +$ $+ 5.4 \cdot 10^{-7} G^6_{-32} + 5.3 \cdot 10^{-8} G^7_{-34}$ $\phi_4 = 0.4I$

Unlike many other glucose-insulin interaction models, MP systems do not use differential equations and in principle can be efficiently implemented in hardware by using only basic mathematical operations.

The unified IVGTT system has pipelined implementation with ten trigger separated stages, as shown in Fig. 3.8. Dependent on the demanded MP system the current or delayed values of glucose must be pushed to the inputs of general MP system. The multiplexers switch proper signals from current to the next stage. The disabling of a branch is implemented using multiplication by zero on the end of that branch. Single glucose/insulin pair is computed in 10 clock cycles. Therefore, at the beginning during first 10 cycles interleaving input data from 10 separate sources allows to fill unified MP system with useful initial samples. On the last stage the results are stored in shift registers with a depth of 10 in order to load value computed at previous iteration at an exact moment when the signals of the right source appear on the inputs.

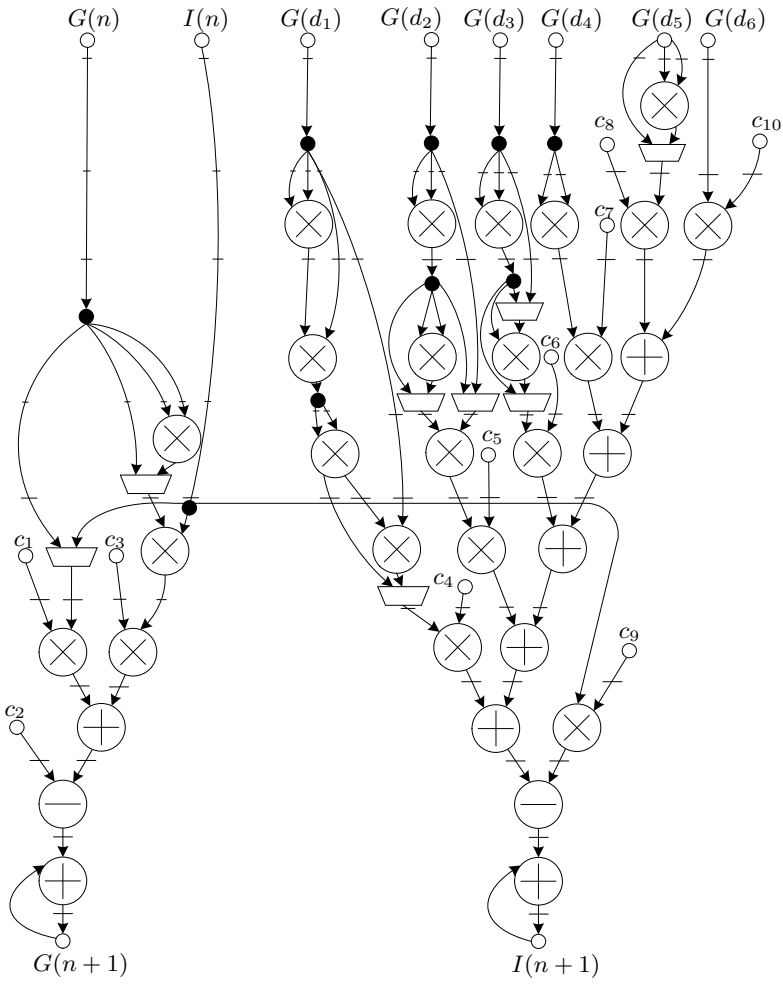


Fig. 3.8. Unified pipelined intra-venous glucose tolerance test metabolic P system implementation

The unified IVGTT system combinative implementation has a similar structure, as shown in Fig. 3.9. The major different compared to the unified pipelined implementation is the lack of trigger separated stages. In case of combinative implementation, the final calculation result is obtained after each cycle and there is no need to use ten separate stages. However, this means that the complex scheme needs to be calculated in one step, potentially decreasing maximum frequency.

There are also many similarities between the two previously discussed unified implementation techniques. As shown in Fig. 3.8 and Fig. 3.9, the input signals and the number of multiplexers is the same.

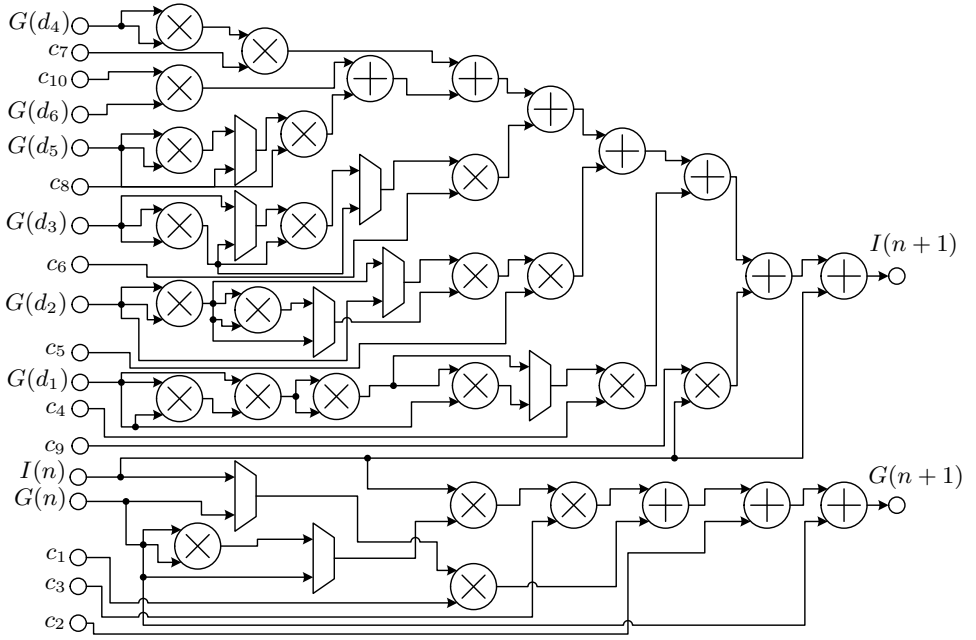


Fig. 3.9. Unified combinative intravenous glucose tolerance test metabolic P system implementation

3.3. Conclusions of the Third Chapter

After presenting the common and original techniques for Metabolic P system implementation in FPGA, the following conclusions can be made:

1. The proposed new data structure of JSON format can be used to represent MP systems and assist in automated implementation in FPGA.
2. The three common techniques for MP system implementation are combinative, single DSP slice and pipelined. They are used for implementing a single MP system in FPGA.
3. Unification should be used to efficiently implement multiple MP systems of the same type. The novel unified implementation technique allows to calculate several MP systems using one FPGA implementation.
4. A set of six different IVGTT MP systems implemented using the unified pipelined technique is calculated in 10 stages. The desired MP system type can be selected by an external signal.

4

Evaluation of Metabolic P System Implementation in Hardware

In the following chapter the developed MP system implementation in FPGA techniques are evaluated by implementing real MP systems. The selected bioengineering dataset of IVGTT MP systems, previously presented in Section 3.2.2, is used to evaluate the implementation techniques. To perform the evaluation, an experimental electronic system based on Xilinx FPGA development board is created.

The investigation procedure is presented in Section 4.1. In Section 4.2 implementation of a single MP system is evaluated. Single MP system implementation is then used to evaluate calculation accuracy in Section 4.2.1 and implementation quality of different common techniques in Section 4.2.2. Using the results of calculation accuracy investigation, a suitable word length is selected for further multiple MP system investigation. The quality of multiple MP system implementation using the proposed unified implementation technique is evaluated in Section 4.3. The different variations of the developed technique are also compared. Complete implementation quality is evaluated using the proposed MP system implementation in FPGA quality criteria and visual representation using a spider chart.

The research results are published in author publications (Kulakovskis, Navakauskas 2016; Kulakovskis *et al.* 2016; Kulakovskis 2019). The main results are announced in international AIEEE (Vilnius, 2016) and national "Science – Future of Lithuania" (Vilnius, 2017), PhD Week (Kaunas, 2018) scientific conferences.

4.1. Investigation Procedure

The evaluation of MP system implementation in FPGA is performed in two major steps: by implementing a single MP system and by implementing multiple MP systems. The IVGTT MP system is selected for implementation. The dataset of six IVGTT system types is used for the evaluation. Single MP system is implemented using common techniques and multiple MP systems are implemented by using the proposed unified implementation technique.

4.1.1. Controllable Parameters

A fixed point number has a certain number of bits allocated to the integer and fractional parts. This balance as well as the total word length is important for the accuracy of calculations. The word length including the number of bits allocated to the integer and fractional parts is a controlled parameter used to determine the optimal balance between accuracy and other quality criteria.

The type of implemented MP system is also a controllable parameter. The IVGTT dataset contains six different MP systems that require different amounts of resources when implemented in FPGA as a single system. The impact of using fixed word length also depends on the implemented IVGTT system type.

Another controlled parameter is the number of simultaneous MP systems implemented using the developed technique. This is used to assess the scalability of the complete system and determine how well the technique is suited for multiple MP system implementation.

Along with the number of implemented MP systems, the capacity of the FPGA is also controlled by selecting the appropriate FPGA chip model. The experiments were performed using Xilinx FPGA development board with Zynq-7000 XC7Z020-1 chip, that in total has 220 DSP slices and $53,200$ LUT. The maximum clock speed and maximum power consumption are 200 MHz and 4.9 W respectively[1].

4.1.2. Assessment Parameters

The implemented systems are evaluated according to calculation accuracy that is measured using two different parameters: MAE and RMSE. Although RMSE is used when evaluating complete system quality, MAE is useful as an additional criteria to better highlight the differences of using various word lengths.

The MP system implementations are also evaluated according to FPGA resource usage and result acquisition speed. These parameters determine how effectively FPGA resources are being utilized. Lower resource usage enables the use of smaller FPGA

[1] Xilinx Inc., Zynq-7000 All Programmable SoC: DC and AC Switching Characteristics (15 June 2017). https://www.xilinx.com/support/documentation/data_sheets/ds187-XC7Z010-XC7Z020-Data-Sheet.pdf

and increases the affordability of the complete system. Higher maximum frequency value allows for power saving measures such as sleep intervals as well as enabling multiple MP system computation using separate time slots.

When evaluating FPGA resource usage, the number of used LUT and DSP elements is acquired from the Xilinx software after the VHDL compile procedure. Number of used IOB is also acquired from the same software and is used when estimating the interface complexity.

The results acquisition speed or throughput is acquired by using two parameters: maximum frequency provided by Xilinx software and latency that depends on the used implementation technique and selected MP system type.

All these assessment parameters are used to calculate the complete system implementation quality. The complete quality consists of five different qualities that can be evaluated separately.

4.1.3. Procedural Steps

First, the implementation of single IVGTT MP system is investigated. The main objective of this investigation is to determine the calculation accuracy of when using fixed word length and FPGA resource usage of common implementation techniques.

A reference point is needed to compare the accuracy of the FPGA implementation. For this purpose an equivalent implementation in MATLAB™ software is used. The reference implementation uses double precision floating point calculations to model the MP systems. These results are then used to determine the RMSE and MAE.

The use of fixed point calculations has a meaningful impact on the calculation accuracy. Therefore this aspect must be investigated to determine and select adequate word length to achieve sufficient accuracy while using the least possible amount of FPGA resources.

Usage of FPGA resources is then estimated using the selected word length that is suitable for all IVGTT system types. Three different common implementation techniques, presented in Chapter 3, are used to implement all of the IVGTT systems.

After investigating single IVGTT MP system implementation accuracy and resource usage parameters, the same approach is applied to multiple unified MP system implementation.

The developed unified MP system implementation technique allows for the implemented system fluxes to be dynamically reconfigured. This enables the application of the IVGTT system implementation for multiple patients at the same time. The resource usage is investigated using unified pipelined and unified combinative IVGTT system implementations.

Finally, the complete prototype of IVGTT MP system implementation using the developed technique is investigated to determine the complete system quality and select the most efficient implementation technique. The impact of multiple simultaneous unified MP system calculations on the FPGA resource usage is then evaluated.

4.1.4. Risk Control

It is important to keep in mind the properties of used FPGA when determining the resource usage parameters. Each FPGA has a certain number of DSP and LUT components that can not be exceeded. This limitation can have an effect on the investigation results because a real FPGA is used to implement the IVGTT MP system. This means that the implementation complexity is limited by the physical FPGA resources.

The amount of required FPGA resources is significantly increased when using long binary word length for high precision calculations. Therefore when investigating the calculation accuracy of various implementations a sifficiently large FPGA chip has to be selected for the results to be reliable.

4.2. Evaluation of Single Metabolic P System Implementation

Single MP system implementation evaluation consists of two parts. First, calculation accuracy is investigated. The order of calculations of IVGTT systems is always the same no matter what implementation technique is used. Therefore it is enough to investigate the accuracy in case where only one MP system is implemented. Secondly, the resource usage of a single IVGTT system implemented using different common techniques is investigated. As the structure and complexity of all types of IVGTT systems is different, it is important to investigate each of them separately. Let's proceed to the calculation accuracy investigation.

4.2.1. Calculation Accuracy Investigation

The calculation accuracy depends on the used word length and the complexity of the selected MP system. The results of the calculation accuracy can be applied to all different implementation techniques. When implementing IVGTT MP systems the calculation order was the same across the different implementation techniques. This allows to determine the calculation error irrespective of the used implementation technique. Therefore the following accuracy results are shown only in relation to the selected MP system and the binary word length.

Calculated output signals of the selected IVGTT MP systems from Table 3.1 are shown in Fig. 4.1 for insulin values and Fig. 4.2 for glucose values. These results are calculated using floating point arithmetic simulated in MATLAB™ software. The different complexity of signals, as well as different magnitudes, can have significant impact on the calculation accuracy when implemented in FPGA with limited word length. Therefore the accuracy is investigated separately for each MP system.

When comparing output signals of insulin and glucose it can be seen that there is more signal variance in the insulin output. This means that a higher error can be

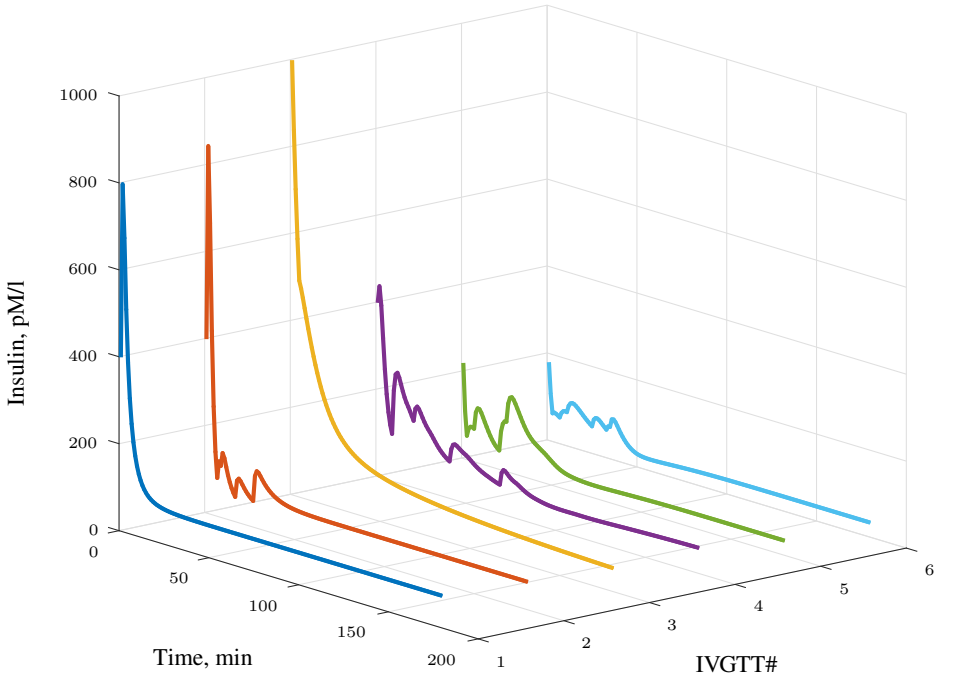


Fig. 4.1. Insulin output signal of IVGTT1–IVGTT6 systems from Table 3.1 implementations

expected in insulin output signal compared to glucose signal when implemented in FPGA using fixed word length. Also it can be noted that the IVGTT1 system has the least amount of signal variance and the smoothest signal curve which should improve FPGA implementation accuracy.

The word lengths selected for the investigation were 18, 24, 32, 40, 48 b. The shortest length of 18 was selected as it is the only possible length that can be used in single DSP technique. The other word lengths can be used in combinative, pipelined and unified techniques.

According to the E_{RMSE} and E_{MAE} values in Table 4.1, the complexity of the IVGTT equations greatly affects the accuracy of the results. The acceptable normalized RMSE is considered to be not more than 15 %, and the acceptable normalized MAE no more than 10 %. At least 32 b word length must be used to achieve this target for all implemented IVGTT system types, although 18 b is enough for IVGTT1 and IVGTT2 and 24 b is enough for IVGTT5 and IVGTT6. Table cells where error is acceptable are marked in blue and the values with minimal word length are highlighted.

Also it is important to note that IVGTT1 and IVGTT2 systems consist of equations that require less bits to accurately implement in fixed point arithmetic. As a result it

Table 4.1. Calculation error of insulin (I) and glucose (G) in intra-venous glucose tolerance test metabolic P systems implemented using different word lengths. Root mean square error of less than 15 % and mean absolute error of less than 10 % are highlighted

Type	Metric	18 b		24 b		32 b		40 b		48 b	
		I	G	I	G	I	G	I	G	I	G
IVGTT1	E_{RMSE}	0.04	0.01	0.01	0.01	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-7}	10^{-7}
	E_{MAE}	0.01	0.01	10^{-3}	0.01	10^{-3}	0.01	10^{-4}	10^{-3}	10^{-8}	10^{-7}
IVGTT2	E_{RMSE}	0.10	0.03	0.10	0.01	0.10	0.01	0.07	10^{-3}	10^{-3}	10^{-3}
	E_{MAE}	0.02	0.01	0.02	0.03	0.02	10^{-3}	0.01	0.01	10^{-4}	10^{-3}
IVGTT3	E_{RMSE}	0.38	0.87	0.26	0.63	10^{-3}	0.01	10^{-3}	0.01	10^{-7}	10^{-7}
	E_{MAE}	0.37	0.85	0.25	0.62	10^{-3}	0.01	10^{-5}	10^{-4}	10^{-8}	10^{-7}
IVGTT4	E_{RMSE}	0.27	0.89	0.27	0.89	0.13	0.10	0.13	0.10	0.10	0.06
	E_{MAE}	0.25	0.87	0.25	0.87	0.06	0.08	0.06	0.08	0.05	0.03
IVGTT5	E_{RMSE}	0.42	0.90	0.08	0.10	0.07	0.03	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	E_{MAE}	0.40	0.88	0.06	0.06	0.04	0.03	10^{-3}	10^{-3}	10^{-5}	10^{-6}
IVGTT6	E_{RMSE}	0.26	0.89	0.07	0.08	0.05	0.02	0.02	10^{-3}	0.01	10^{-3}
	E_{MAE}	0.25	0.88	0.05	0.07	0.02	0.01	10^{-3}	10^{-3}	10^{-5}	10^{-5}

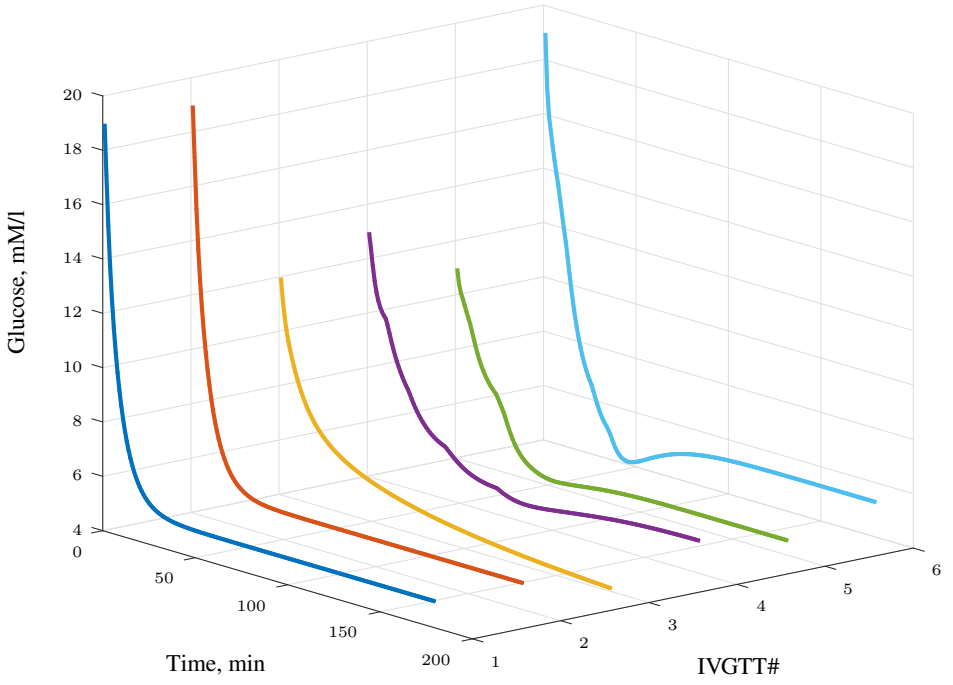


Fig. 4.2. Glucose output signal of IVGTT1–IVGTT6 systems from Table 3.1 implementations

can be seen that in the case where only these systems would be used, 18 b word length that is used in single DSP implementation technique would be sufficient to achieve the acceptable error value. Thus the single DSP technique will only be used to implement IVGTT1 and IVGTT2 in the following investigation.

The error values shown in Table 4.1 are averaged over all sample values. This allows to measure accuracy using a single value, but in the case of IVGTT implementations the errors are usually not uniform across all values. This is shown in the output signals of insulin in Fig. 4.3 and glucose in Fig. 4.4 of IVGTT5 system implemented using various word lengths. The 18 b implementation is omitted from comparison because its output signal is very inaccurate and does not even follow a similar path. The 48 b implementation is also omitted as its error is very low and can not be clearly seen in a graphical representation. As can be seen from the rest of the output signals, although 24 b implementation accuracy is still acceptable, there are some periods where the error is considerable and others where the error is negligible. This shows that higher word length is advantageous for implementation even if the error value is considered acceptable with lower word lengths.

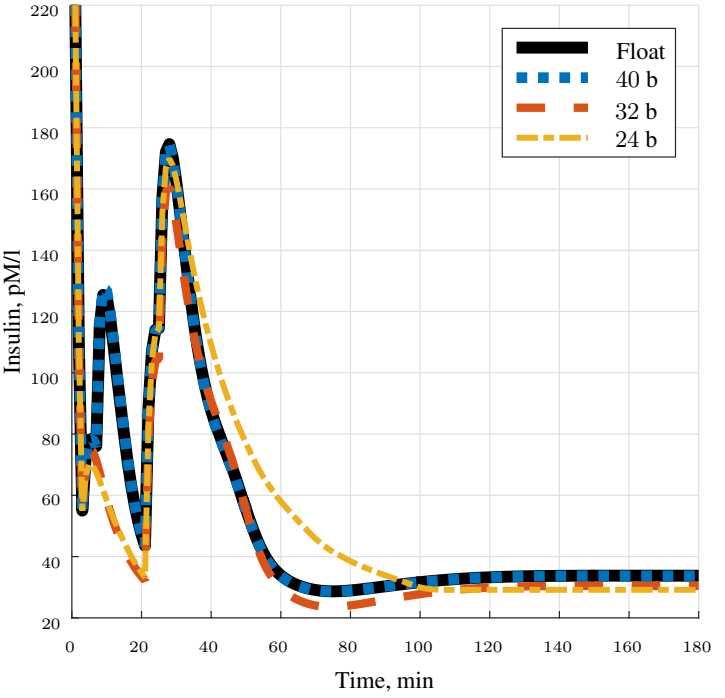


Fig. 4.3. Insulin output signal of fixed point implementations of IVGTT5 system using different word lengths compared to reference floating point implementation

4.2.2. Results of Single Metabolic P System Implementation Quality Evaluation

The resource usage of the single DSP implementation is shown in the first part of Table 4.2. All single DSP implementations use word length of 18 b, that is limited by the architecture of the DSP cells. Because of the word length limitation, the single DSP implementation is only suitable for IVGTT1 and IVGTT2 systems. As can be seen in Table 4.1, the calculation accuracy of other systems is too low when only 18 b word length is used.

The single DSP implementation has a relatively high latency. This is because all calculations are performed by one DSP cell. On the other hand, the relatively short signal path means that high maximum frequency can be achieved. In this case $f_{\max} = 194$ MHz for all implemented MP systems.

As the single DSP implementation technique has only limited suitability for IVGTT application (only 2 out of 6 systems can be accurately implemented), the two other discussed methods are used for the following investigation. The 32 b word length is se-

Table 4.2. Detailed field programmable gate array resource consumption of intra-venous glucose tolerance test metabolic P systems implemented using different techniques

Word	Type	Technique	$f_{\max}$, MHz	DSP	LUT	Latency	$N_{\equiv \text{LUT}}$	f_{val}, MHz
18 b	IVGTT1	Single DSP	194	1	211	17	407	11
	IVGTT2	Single DSP	194	1	772	29	968	7
32 b	IVGTT1	Pipelined	66	14	965	6	3709	11
		Combinative	22	14	1044	1	3788	22
	IVGTT2	Pipelined	65	26	1982	10	7078	7
		Combinative	29	16	1553	1	4689	29
	IVGTT3	Pipelined	68	12	924	4	3276	17
		Combinative	35	12	938	1	3290	35
	IVGTT4	Pipelined	65	30	1713	8	7593	8
		Combinative	25	20	1420	1	5340	25
	IVGTT5	Pipelined	65	36	2099	8	9155	8
		Combinative	21	26	1667	1	6763	21
	IVGTT6	Pipelined	65	50	2669	8	12469	8
		Combinative	26	26	1872	1	6968	26

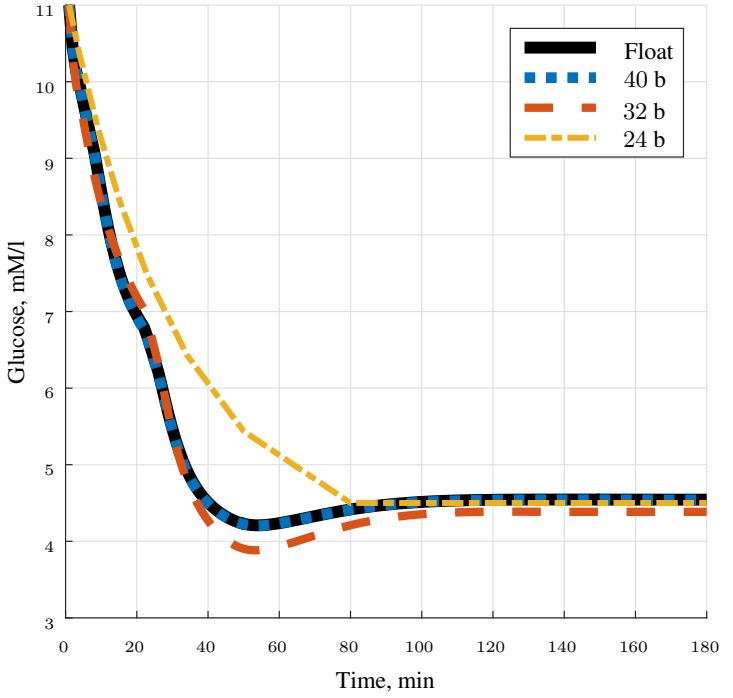


Fig. 4.4. Glucose output signal of fixed point implementations of IVGTT5 system using different word lengths compared to reference floating point implementation

lected because it is sufficient to achieve required accuracy for all implemented IVGTT systems.

The resource usage results of pipelined implementation are shown in the second part of Table 4.2. IVGTT6 system requires the most DSP and LUT resources when compared to other systems. IVGTT3 system uses the least amount of resources to implement. This system also requires the least amount of clock cycles to calculate in a pipelined way. On the other hand the IVGTT2 system has the highest latency of 10 clock cycles. The maximum frequency is only marginally different when comparing the different MP system implementations. Therefore it can be said that IVGTT2 system is the most challenging to implement from the timing perspective and IVGTT6 system is the most challenging from the FPGA resource consumption perspective.

The results of combinative implementation in Table 4.2 show some key differences from the pipelined implementation. The resource consumption varies similarly to the pipelined implementation, with IVGTT6 system using the most LUT and DSP (together with IVGTT5) resources. IVGTT3 system also uses the least amount of FPGA resources. The bigger difference when comparing with pipelined imple-

mentation is in the frequency and latency metrics. The latency of combinative implementation always equals one clock cycle because calculations are performed over one clock cycle using one long logic chain. This also means that the frequency of the implementation is lower by a factor from 2 to 3 times when compared to pipelined implementation.

To compare pipelined and combinative implementation resource usage, two values are calculated and shown in Table 4.2. The value generation speed (f_{val}) is calculated using Equation 2.15 and the total used equivalent LUT resources ($N_{\equiv\text{LUT}}$) are calculated using Equation 2.17. In the results table the better values (higher speed and lower LUT usage) are highlighted.

According to calculated value generation frequency, combinative implementation achieves higher speed for all implemented MP systems. Although this implementation technique results in lower maximum frequency, the advantage of requiring only one clock cycle to get the results outweighs this.

The more effective implementation technique according to used FPGA resources is not the same for all implemented MP systems. However, 4 out of 6 times the combinative implementation was more effective at resource usage. In the two cases where pipelined implementation was more effective, the differences in used resources were only 2.1 % (IVGTT1) and 0.4 % (IVGTT3).

4.3. Evaluation of Multiple Metabolic P Systems Implementation

Multiple MP system implementation evaluation consists of three parts. First, the detailed implementation quality criteria such as FPGA resource consumption are presented. Secondly, complete implementation quality is investigated by applying MP system quality criteria presented in Chapter 2. Finally, the possibility of implementing multiple unified MP systems at the same time in different FPGA models is investigated.

4.3.1. Results of Unified Implementation Evaluation

The implementations of a single MP system discussed in the previous section can be used for implementing one IVGTT system, but they are not effective in case when multiple IVGTT systems need to be calculated simultaneously. When six different IVGTT systems need to be implemented, six single MP implementations need to fit in the FPGA.

Unified MP system implementations allow the FPGA to contain one implementation that is able to calculate different kinds of IVGTT systems. Two different implementation techniques – pipelined and combinative – are adapted to unified implementation.

Table 4.3. Detailed field programmable gate array resource consumption of intra-venous glucose tolerance test metabolic P systems implemented using unified pipelined and unified combinative techniques

Word	Technique	f_{max}, MHz	DSP	LUT	Latency	IOB	P, mW	$N_{\equiv LUT}$	f_{val}, MHz
18 b	Unified Pipelined	145	6	665	10	78	140	1841	15
	Unified Combinative	47	6	619	1	78	113	1795	47
24 b	Unified Pipelined	114	12	880	10	102	143	3232	11
	Unified Combinative	40	12	818	1	102	113	3170	40
32 b	Unified Pipelined	65	84	4302	10	134	313	20766	7
	Unified Combinative	13	84	4317	1	134	113	20781	13
40 b	Unified Pipelined	59	172	6005	10	166	384	39717	6
	Unified Combinative	11	172	6312	1	166	113	40024	11
48 b	Unified Pipelined	56	220	12636	10	198	489	55756	6
	Unified Combinative	10	220	12705	1	198	113	55825	10
64 b	Unified Pipelined	53	216	27207	10	262	587	69543	5
	Unified Combinative	10	216	25838	1	262	113	68174	10

The resource consumption of unified pipelined implementation is presented in Table 4.3. The implementations with lower total number of bits (18 and 24) use significantly less resources (up to 14 times less DSP and 6 times less LUT) when compared to 32 b implementation. However, shorter word length can not be used to accurately calculate all IVGTT systems as shown in the previous section. It can be seen that lower word lengths can not even fully represent all integer values.

Similarly results of unified combinative implementation are also presented in Table 4.3. When compared to unified pipelined implementation, it can be seen that the resource consumption is very similar. The number of DSP cells used is the same for all word lengths and the number of LUT differs insignificantly. This differs from the single MP system implementations, where combinative implementations used less resources most of the time.

The maximum frequency achieved by the unified implementations varies significantly between different word lengths. When looking at 32 b word length implementation, the unified pipelined implementation has 5 times higher maximum frequency. However, in total the unified combinative implementation has higher value generation frequency, as it has latency of only 1 clock cycle, compared to 10 clock cycles required for the unified pipelined implementation.

The number of used IOB is the same for both unified implementations. It only depends on the used word length, as more buffers are needed to output higher precision values.

The power consumption of unified combinative implementation is equal to 113 mW for all word lengths and is from 1.2 to 5.2 times lower than in unified pipelined implementation.

4.3.2. Results of Complete System Quality Evaluation

Total MP system implementation quality was calculated to compare the unified implementation techniques and select a suitable implementation for the complete system. The Q_{MP} value is calculated according to (2.21) and σ_{MP} value is calculated according to (2.25). The quality limiting parameters are based on the selected FPGA chip.

The calculated quality values are shown in Table 4.4. Accuracy $Q_{\mathrm{A}}^{\mathrm{MP}}$ is the same for both unified pipelined and combinative implementations as they both use the same 32 b word length and calculation order. Similarly interface complexity $Q_{\mathrm{I}}^{\mathrm{MP}}$ is also the same as the output signals do not differ between implementations. The unified pipelined implementation has a very slight advantage in resource usage $Q_{\mathrm{R}}^{\mathrm{MP}}$ but the unified combinative implementation has advantages in throughput $Q_{\mathrm{T}}^{\mathrm{MP}}$ and power consumption $Q_{\mathrm{P}}^{\mathrm{MP}}$.

In total the unified combinative implementation has better complete system quality Q_{MP}. This is mostly because of higher throughput as can be seen from quality ratios in Table 4.4. The quality dispersion is also lower for unified combinative implementation. That means the components of complete quality are more evenly placed, as illustrated in Fig. 4.5.

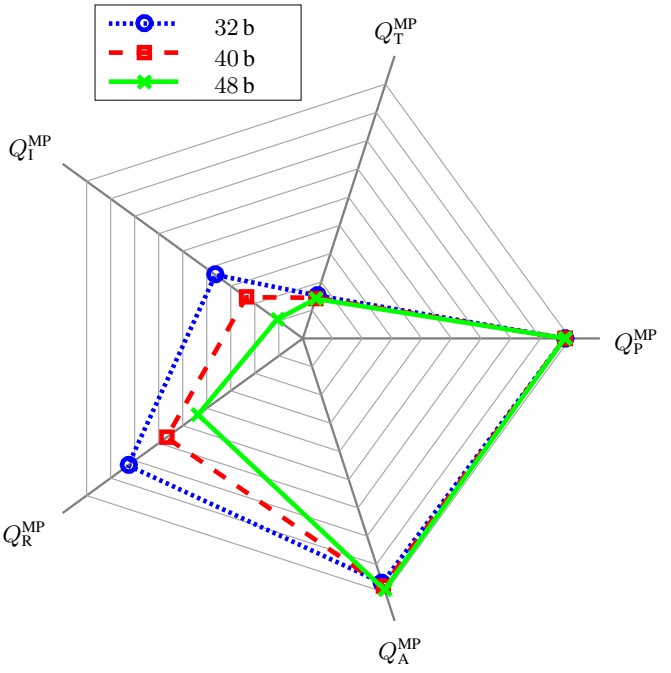


Fig. 4.5. Quality comparison of unified combinative implementation using different word lengths

The change of quality when longer word length is selected is shown in Fig. 4.5. Unified combinative implementation technique is used as it was previously determined to have higher total quality. The change in word length greatly affects resource usage and interface complexity. The gain in accuracy is minimal, as shown in Table 4.1.

4.3.3. Results of Multiple Simultaneous Metabolic P System Implementation

In the previous section the unified combinative implementation was determined to have the best complete system quality for IVGTT MP system. The remaining question now is what size of FPGA chip is needed for multiple MP system implementation using the unified combinative technique.

The previous experiments were performed using the Xilinx FPGA development board with FPGA equivalent to the Artix-7 chip family. It was able to fit the modeled systems, but only one was implemented at a time. Although the main advantage of the unified implementation technique is that multiple types of MP systems can be

Table 4.4. Complete quality comparison of 32 b unified intra-venous glucose tolerance test metabolic P system implementations

Metric	Unified Pipelined	Unified Combinative	Ratio
$Q_{\mathrm{A}}^{\mathrm{MP}}$	95.74	95.74	1
$Q_{\mathrm{T}}^{\mathrm{MP}}$	3.25	6.50	0.5
$Q_{\mathrm{R}}^{\mathrm{MP}}$	78.44	78.42	1
$Q_{\mathrm{P}}^{\mathrm{MP}}$	93.61	97.69	0.96
$Q_{\mathrm{I}}^{\mathrm{MP}}$	33.00	33.00	1
Q_{MP}	37.62	43.59	0.86
σ_{MP}	43.32	40.86	1.06

modeled by the same circuit, it can calculate different MP types only at separate time slots. There could be use cases where very large number of MP systems need to be calculated simultaneously, for example multiple IVGTT systems for multiple patients in a hospital environment.

The Xilinx 7 series chip family has multiple variants of chips available[2]. The main difference is the number of components, such as DSP and LUT, that determine how many unified IVGTT systems can be run at the same time. It is possible to determine the resources needed for implementing IVGTT systems in other Xilinx 7 series chips, as the previous experiment data is obtained from the same FPGA family. The results for other chip families should be very similar, as long as they have comparable number of DSP and LUT elements.

The smallest FPGA of the Spartan-7 and Artix-7 families can not be used to implement the unified IVGTT system with 32 b precision because it requires more DSP slices than are available in those chips, as shown in Table 4.5. The results also show that it is best to use Spartan-7 XC7S50 or Artix-7 XC7A35T for implementing 1 MP system and Artix-7 XC7A75T for 2 MP systems, because the one step bigger chips can fit the same amount of MP systems and are more expensive.

The results in Table 4.5 show that if high number of unified IVGTT systems needs to be calculated simultaneously, a higher capacity FPGA, such as Virtex-7 XC7VX690T with maximum number of 42 simultaneous MP systems, should be used. However, if high throughput is not required, a cheaper FPGA that can fit one unified IVGTT system can be used to calculate all types of IVGTT systems sequentially, as the unified implementation technique allows dynamic reconfiguration.

Considering practical use of IVGTT systems for glucose monitoring, the total number of systems possible to be calculated by a single FPGA can be defined as:

$$N_{\mathrm{IVGTT}} \triangleq \frac{f_{\mathrm{val}}}{N_{\mathrm{val}}} \times N_{\mathrm{MP}} \times t_{\mathrm{mon}}\,, \tag{4.1}$$

[2]Xilinx Inc., 7 Series FPGAs Overview (ver2.6) (28 February 2018). https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

Table 4.5. Maximum number of 32 b unified intra-venous glucose tolerance test metabolic P systems that fit in different chips of Xilinx 7 series field programmable gate array family

FPGA		DSP		LUT		MP
Family	Chip number	Total	Used	Total	Used	Nmb.
Spartan-7	XC7S6	10	840.0 %	6,000	72.0 %	0
	XC7S15	20	420.0 %	12,800	33.7 %	0
	XC7S25	80	105.0 %	23,360	18.5 %	0
	XC7S50	120	70.0 %	52,160	8.3 %	1
	XC7S75	140	60.0 %	76,800	5.6 %	1
	XC7S100	160	52.5 %	102,400	4.2 %	1
Artix-7	XC7A12T	40	210.0 %	12,800	33.7 %	0
	XC7A15T	45	186.7 %	16,640	25.9 %	0
	XC7A25T	80	105.0 %	23,360	18.5 %	0
	XC7A35T	90	93.3 %	33,280	13.0 %	1
	XC7A50T	120	70.0 %	52,160	8.3 %	1
	XC7A75T	180	46.7 %	75,520	5.7 %	2
	XC7A100T	240	35.0 %	101,440	4.3 %	2
	XC7A200T	740	11.4 %	215,360	2.0 %	8
Kintex-7	XC7K70T	240	35.0 %	65,600	6.6 %	2
	XC7K160T	600	14.0 %	162,240	2.7 %	7
	XC7K325T	840	10.0 %	326,080	1.3 %	10
	XC7K355T	1,440	5.8 %	356,160	1.2 %	17
	XC7K410T	1,540	5.5 %	406,720	1.1 %	18
	XC7K420T	1,680	5.0 %	416,960	1.0 %	20
	XC7K480T	1,920	4.4 %	477,760	0.9 %	22
Virtex-7	XC7VX330T	1,120	7.5 %	326,400	1.3 %	13
	XC7VX415T	2,160	3.9 %	412,160	1.0 %	25
	XC7VX485T	2,800	3.0 %	485,760	0.9 %	33
	XC7VX550T	2,880	2.9 %	554,240	0.8 %	34
	XC7VX690T	3,600	2.3 %	693,120	0.6 %	42
	XC7VX980T	3,600	2.3 %	979,200	0.4 %	42
	XC7VX1140T	3,360	2.5 %	1,139,200	0.4 %	40

where N_{IVGTT} – total number of IVGTT systems calculated, f_{val} – value generation frequency, N_{val} – total number of values in discrete time, N_{MP} – maximum number of MP systems calculated by one FPGA, t_{mon} – glucose-insulin monitoring sensor reading interval.

The f_{val} of 32 b unified combinative implementation is 13 MHz and the N_{val} of IVGTT systems is 200. The Virtex-7 XC7VX690T FPGA can calculate 42 unified IVGTT systems simultaneously. Glucose monitoring sensors usually produce aver-

aged readings every 3–5 minutes (McAdams, Rizvi 2016) and the glucose signal is acquired with the interval in the order of seconds (Olczuk, Priefer 2018). For the following calculation the interval of 1 s will be used. Accordint to this data the total number of IVGTT systems calculated in one FPGA is:

$$N_{\text{IVGTT}} = \frac{13 \cdot 10^6}{200} \times 42 \times 1 = 2.73 \cdot 10^6 \,. \tag{4.2}$$

The results in (4.2) show that it is possible to calculate more than 2 million IVGTT systems simultaneously when using a single powerful FPGA chip. Of course, this doesn't take into account other functions that would have to be performed by the FPGA in a real-world implementation, such as glucose sensor signal processing or visual display or results. However, the fact that MP system implementation effectively uses FPGA resources by requiring only a small interval for one system calculation, shows that this implementation can be used in affordable portable low power electronic devices.

4.4. Conclusions of the Fourth Chapter

1. The word lengths of at least 18 b for IVGTT-1 and IVGTT-2, 24 b for IVGTT-5 and IVGTT-6, and 32 b for IVGTT-3 and IVGTT-4 MP systems should be used to achieve RMSE of less than 15 %.
2. Combinative implementation technique achieves higher value generation frequency for all investigated IVGTT MP systems and uses less FPGA resources in 4 out of 6 cases when compared to pipelined implementation.
3. The proposed unified combinative IVGTT MP system implementation technique achieves higher value generation frequency and the proposed unified pipelined technique achieves lower power consumption with all investigated word lengths.
4. The proposed unified combinative IVGTT MP system implementation has 1.16 times higher average quality and 1.06 times lower quality dispersion when compared to unified pipelined implementation.
5. The qualities of resource usage and interface complexity decrease at a higher rate than the quality of accuracy increases when higher than 32 b word lengths are used with unified combinative implementation.
6. Multiple unified IVGTT MP systems with 32 b word length can be implemented in a single Xilinx 7 series family FPGA, with Virtex-7 XC7VX690T chip being able to fit up to 42 unified systems and in practice calculate up to $2.73 \cdot 10^6$ IVGTT systems simultaneously.

General Conclusions

The problem of hardware implementation of the metabolic P system mathematical model has been solved. The following significant results for scientific field of Electrical and Electronic engineering are obtained:

1. The presented metabolic P system implementation techniques can be applied to implement all six known and investigated intra-venous glucose tolerance test metabolic P systems in field programmable gate arrays with root mean square error not higher than 15% using word length of at least 32 bits.
2. The new metabolic P system combined quality metric and its graphic representation clearly separates the quality of different metabolic P system implementations in field programmable gate arrays and allows to analyze quality characteristics appropriate for these systems.
3. The offered unified combinative intra-venous glucose tolerance test metabolic P system implementation technique ensures 2 to 3 times higher speed compared to the unified pipelined implementation technique.

In the future the work presented in this thesis can be continued by expanding the variety of modeled IVGTT systems with real experimental medical data and testing the created electronic system in real word applications. The developed techniques can also be applied to model new types of metabolic P systems, helping to solve different kinds of problems.

References

Abdelouahab, K.; Bourrasset, C.; Pelcat, M.; Berry, F.; Quinton, J.-C.; Serot, J. 2016. A holistic approach for optimizing DSP block utilization of a CNN implementation on FPGA, in *Proceedings of the 10th International Conference on Distributed Smart Camera*, 69–75. [see 1 p.]

Ackerman, E.; Gatewood, L.; Rosevear, J.; Molnar, G. 1965. Model studies of blood-glucose regulation, *Bulletin of Mathematical Biology* 27: 21–37. [see 28 p.]

Alhazov, A.; Leporati, A.; Mauri, G.; Porreca, A. E.; Zandron, C. 2014. Space complexity equivalence of P systems with active membranes and turing machines, *Theoretical Computer Science* 529: 69–81. [see 10 p.]

American Diabetes Association 2014. Standards of medical care in diabetes – 2014, *Diabetes Care* 37(Supplement 1): S14–S80. [see 2 p.]

Arroyo, F.; Baranda, A.; Castellanos, J.; Paun, G. 2002. Membrane computing: The power of (rule) creation, *Journal of Universal Computer Science* 8(3): 369–381. [see 11 p.]

Bao, B.; Mitrea, C.; Wijesinghe, P.; Marchetti, L.; Girsch, E.; Farr, R. L.; Boerner, J. L.; Mohammad, R.; Dyson, G.; Terlecky, S. R.; Bollig-Fischer, A. 2017. Treating triple negative breast cancer cells with erlotinib plus a select antioxidant overcomes drug resistance by targeting cancer cell heterogeneity, *Scientific Reports* 7: 44 125. [see 19 p.]

Bekiari, E.; Kitsios, K.; Thabit, H.; Tauschmann, M.; Athanasiadou, E.; Karagiannis, T.; Haidich, A.-B.; Hovorka, R.; Tsapas, A. 2018. Artificial pancreas treatment for outpatients with type 1 diabetes: Systematic review and meta-analysis, *BMJ Clinical Research* 361. [see 3 p.]

Bernardini, F.; Gheorghe, M. 2004. Population P systems, *Journal of Universal Computer Science* 10(5): 509–539. [see 11 p.]

Bernardini, F.; Manca, V. 2003a. Dynamical aspects of P systems, *Biosystems* 70(2): 85–93. [see 12, 13, 14 p.]

Bernardini, F.; Manca, V. 2003b. P systems with boundary rules, in *Membrane Computing*, Springer-Verlag, 107–118. [see 12, 13 p.]

Bian, H.; Ling, A. C.; Choong, A.; Zhu, J. 2010. Towards scalable placement for FPGAs, in *FPGA 10*, 147–156. [see 23 p.]

Bianco, L.; Fontana, F.; Franco, G.; Manca, V. 2006a. P systems for biological dynamics, in *Applications of Membrane Computing*, Springer Berlin Heidelberg, 83–128. [see 19 p.]

Bianco, L.; Fontana, F.; Manca, V. 2006b. P systems with reaction maps, *International Journal of Foundations of Computer Science* 27–48. [see 18, 22 p.]

Bianco, L.; Manca, V.; Marchetti, L.; Petterlini, M. 2007. Psim: a simulator for biomolecular dynamics based on P systems, in *IEEE Congress on Evolutionary Computation, CEC 2007*, 883–887. [see 20 p.]

Bianco, L.; Pescini, D.; Siepmann, P.; Krasnogor, N.; Romero-Campero, F. J.; Gheorghe, M. 2006c. Towards a P systems pseudomonas quorum sensing model, in *Membrane Computing*, Springer, 197–214. [see 19 p.]

Bollig-Fischer, A.; Marchetti, L.; Mitrea, C.; Wu, J.; Kruger, A.; Manca, V.; Draghici, S. 2014. Modeling time-dependent transcription effects of HER2 oncogene and discovery of a role for E2F2 in breast cancer cell-matrix adhesion, *Bioinformatics* 30(21): 3036–3043. [see 19 p.]

Brijder, R.; Cavaliere, M.; Riscos-Nunez, A.; Rozenberg, G.; Sburlan, D. 2007. Membrane systems with marked membranes, *Electronic Notes in Theoretical Computer Science* 171(2): 25–36. [see 11 p.]

Brijder, R.; Cavaliere, M.; Riscos-Nunez, A.; Rozenberg, G.; Sburlan, D. 2008. Membrane systems with proteins embedded in membranes, *Theoretical Computer Science* 404(1–2): 26–39. [see 11 p.]

Byrne, M. 2016. *Intel Bets $16.7 Billion on the Massively Parallel Future of Computing* [interactive] [1 January 2016]. Available online at: https://motherboard.vice.com/en_us/article/nz7vmd/intel-bets-167-billion-on-the-massively-parallel-future. [see 1 p.]

Castellini, A.; Franco, G.; Manca, V. 2009. Toward a representation of hybrid functional petri nets by MP systems, in *Natural computing*, Springer, 28–37. [see 19 p.]

Castellini, A.; Franco, G.; Pagliarini, R. 2011a. Data analysis pipeline from laboratory to MP models, *Natural Computing* 10: 55–76. [see 19 p.]

Castellini, A.; Franco, G.; Pagliarini, R. 2011b. *NPQ phenomenon* [interactive] [October 2015]. Available online at: http://mplab.scienze.univr.it/external/natcomp/NPQ_stepwise_tab4.html. [see 18 p.]

Castellini, A.; Manca, V. 2009. MetaPlab: A computational framework for metabolic P systems, in *Membrane Computing*, Springer Berlin Heidelberg, 157–168. [see 21 p.]

Castellini, A.; Manca, V.; Zucchelli, M. 2015. An evolutionary procedure for inferring MP systems regulation functions of biological networks, *Natural Computing* 14(3): 375–391. [see 19 p.]

Castellini, A.; Paltrinieri, D.; Manca, V. 2014. MP-GeneticSynth: inferring biological network regulations from time series, *Bioinformatics* 31(5): 785–787. [see 21 p.]

Ceterchi, R.; Mutyam, M.; Paun, G.; Subramanian, K. 2003. Array-rewriting P systems, *Natural Computing* 2(3): 229–249. [see 10 p.]

Chai, T.; Draxler, R. R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – arguments against avoiding RMSE in the literature, *Geoscientific model development* 7(3): 1247–1250. [see 24 p.]

Cobelli, C.; Renard, E.; Kovatchev, B. 2011. Artificial pancreas: Past, present, future, *Diabetes* 60(11): 2672–2682. [see 3 p.]

Cong, J.; Jiang, W. 2008. Pattern-based behavior synthesis for FPGA resource reduction, in *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*, 107–116. [see 55 p.]

DARPA 2013. *SyNAPSE Program* [interactive] [11 January 2013]. Available online at: http://www.artificialbrains.com/darpa-synapse-program. [see 1 p.]

Dillien, P. 2017. And the winner of best FPGA of 2016 is…, in *EETimes* [interactive], [3 March 2017]. Available online at: https://www.eetimes.com/author.asp?do_id=1331443. [see 25 p.]

Dutta, S.; Botros, N. M. 2013. FPGA synthesis of glucose-insulin feedback system, in *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV)*, 1–5. [see 28 p.]

Freund, R.; Paun, G.; Perez-Jimenez, M. J. 2005. Tissue P systems with channel states, *Theoretical Computer Science* 330(1): 101–116. [see 11 p.]

Frisco, P. 2009. *Computing with Cells: Advances in Membrane Computing*. Oxford University Press. [see 10 p.]

Frisco, P.; Hoogeboom, H.; Sant, P. 2002. A direct construction of a universal P system, *Annales Societatis Mathematicae Polonae. Series 4: Fundamenta Informaticae* 49(1–3): 103–122. [see 11 p.]

Gharghory, S. M.; El-Dib, D. A. 2016. Fuzzy control system for regulating the blood glucose level of diabetes patients implemented on FPGA, *Journal of Circuits, Systems and Computers* 25(12): 1–17. [see 28 p.]

Ghorbani, M.; Bogdan, P. 2013. A cyber-physical system approach to artificial pancreas design, in *Proceedings of the ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, 1–10. [see 28 p.]

Grewal, G.; Areibi, S.; Westrik, M.; Abuowaimer, Z.; Zhao, B. 2017. Automatic flow selection and quality-of-result estimation for FPGA placement, in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 115–123. [see 23 p.]

Guiraldelli, R. H. G.; Manca, V. 2015a. Automatic translation of $MP^{+}V$ systems to register machines, in *International Conference on Membrane Computing*, Springer, 185–199. [see 2, 22, 48 p.]

Guiraldelli, R. H. G.; Manca, V. 2015b. The computational universality of metabolic computing, *arXiv preprint* . [see 3, 22 p.]

HariKumar, R.; Sudhaman, V.; Babu, C. G. 2012. FPGA synthesis of fuzzy (PD and PID) controller for insulin pumps in diabetes using Cadence, *International Journal of Soft Computing and Engineering* 1(6): 324–331. [see 28 p.]

Haselman, M.; Miyaoka, R.; Lewellen, T. K.; Hauck, S.; McDougald, W.; Dewitt, D. 2009. FPGA-based front-end electronics for positron emission tomography, in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 93–102. [see 24 p.]

HBP 2018. *Human Brain Project* [interactive] [June 2018]. Available online at: http://www.humanbrainproject.eu. [see 1 p.]

Head, T. 1987. Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors, *Bulletin of Mathematical Biology* 49(6): 737–759. [see 11 p.]

Hynes, V. 2013. The trend toward self-diagnosis, *Canadian Medical Association Journal* 185(3): E149–E150. [see 3 p.]

Intel 2018. *Intel FPGA Product Catalog* [interactive] [1 September 2018]. Available online at: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/sg/product-catalog.pdf. [see 25 p.]

Ionescu, M.; Paun, G.; Yokomori, T. 2006. Spiking neural P systems, *Fundamenta Informaticae* 279–308. [see 11 p.]

Jamieson, P.; Luk, W.; Wilton, S. J.; Constantinides, G. A. 2009. An energy and power consumption analysis of FPGA routing architectures, in *Field-Programmable Technology, 2009 International Conference on*, 324–327. [see 24 p.]

Jiang, K.; Song, B.; Shi, X.; Song, T. 2012. An overview of membrane computing, *Journal of Bioinformatics and Intelligent Control* 1(1): 17–26. [see 10 p.]

Khosla, V. 2012. *Technology will replace 80% of what doctors do* [interactive] [4 December 2012]. Available online at: http://fortune.com/2012/12/04/technology-will-replace-80-of-what-doctors-do/. [see 3 p.]

Krishna, S. N. 2005. The power of mobility: Four membranes suffice, in *New Computational Paradigms*, vol. 3526: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 242–251. [see 11 p.]

Krishna, S. N. 2009. Membrane computing with transport and embedded proteins, *Theoretical Computer Science* 410(4–5): 355–375. [see 11 p.]

Krishna, S. N. 2011. An overview of membrane computing, in *Distributed Computing and Internet Technology*, vol. 6536: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1–14. [see 11, 12 p.]

Krishna, S. N.; Rama, R. 1999. A variant of P-systems with active membranes: Solving NP-complete problems, *Romanian Journal of Information Science and Technology* 2(4): 305–394. [see 10 p.]

Krishna, S. N.; Rama, R. 2001. P systems with replicated rewriting, *Journal of Automata, Languages and Combinatorics* 6: 345–350. [see 11 p.]

Li, P.; Yu, L.; Fang, Q.; Lee, S.-Y. 2015. A simplification of cobelli's glucose–insulin model for type 1 diabetes mellitus and its FPGA implementation, *Medical & biological engineering & computing* 1–15. [see 28 p.]

Luenberger, D. 1979. *Introduction to Dynamic Systems. Theory, Models, and Applications*. John Wiley & Sons. [see 16 p.]

Mametjanov, A.; Balaprakash, P.; Choudary, C.; Hovland, P. D.; Wild, S. M.; Sabin, G. 2015. Autotuning FPGA design parameters for performance and power, in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, 84–91. [see 23 p.]

Manca, V. 2009. Fundamentals of metabolic P systems, *Handbook of membrane computing* 19: 489–498. [see 15 p.]

Manca, V. 2013. *Infobiotics: Information in Biotic Systems*. Springer. [see 1, 16, 18 p.]

Manca, V.; Bianco, L. 2008. Biological networks in metabolic P systems, *Biosystems* 91(3): 489–498. [see 19 p.]

Manca, V.; Bianco, L.; Fontana, F. 2005. Evolution and oscillation in P systems: Applications to biological phenomena, in *Membrane Computing*, 63–84. [see 12, 14 p.]

Manca, V.; Marchetti, L. 2009. XML representation of metabolic P systems, in *IEEE Congress on Evolutionary Computation*, 3103–3110. [see 21 p.]

Manca, V.; Marchetti, L. 2010a. Goldbeter's mitotic oscillator entirely modeled by MP systems, in *Membrane computing*, Springer-Verlag Berlin, 273–284. [see 19 p.]

Manca, V.; Marchetti, L. 2010b. Metabolic approximation of real periodical functions, *The Journal of Logic and Algebraic Programming* 79(6): 363–373. [see 15, 16 p.]

Manca, V.; Marchetti, L.; Pagliarini, R. 2011. MP modeling of glucose-insulin interactions in the intravenous glucose tolerance test, *International Journal of Natural Computing Research* 2(3): 13–24. [see 2, 3, 7, 20, 63 p.]

Manca, V.; Pagliarini, R.; Zorzan, S. 2009. A photosynthetic process modelled by a metabolic P system, *Natural Computing* 8(4): 847–864. [see 18 p.]

Marchetti, L.; Manca, V. 2012. A methodology based on MP theory for gene expression analysis, in *Membrane Computing*, vol. 7184, ed. by Gheorghe, M.; Paun, G.; Rozenberg, G.; Salomaa, A.; Verlan, S.: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 300–313. [see 2 p.]

Marchetti, L.; Manca, V. 2014. MpTheory Java library: a multi-platform Java library for systems biology based on the metabolic P theory, *Bioinformatics* 31(8): 1328–1330. [see 21 p.]

Martin-Vide, C.; Paun, G.; Pazos, J.; Rodriguez-Paton, A. 2003. Tissue P systems, *Theoretical Computer Science* 296(2): 295–326. [see 10 p.]

Martin-Vide, C.; Pazos, J.; Paun, G.; Rodriguez-Paton, A. 2002. A new class of symbolic abstract neural nets: Tissue P systems, in *Computing and Combinatorics*, Springer, 290–299. [see 11 p.]

McAdams, B. H.; Rizvi, A. A. 2016. An overview of insulin pumps and glucose sensors for the generalist, *Journal of Clinical Medicine* 5(1): 5. [see 83 p.]

Minsky, M. 1967. *Computation: Finite and Infinite Machines*. Prentice Hall. [see 22 p.]

Nguyen, Q. V.; Caro, A.; Raoux, M.; Quotb, A.; Floderer, J.-B.; Bornat, Y.; Renaud, S.; Lang, J. 2013. A novel bioelectronic glucose sensor to process distinct electrical activities of pancreatic beta-cells, in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 172–175. [see 28 p.]

Nguyen, V.; Kearney, D.; Gioiosa, G. 2007. Balancing performance, flexibility, and scalability in a parallel computing platform for membrane computing applications, in *Membrane Computing: 8th International Workshop*, Springer Berlin Heidelberg, 385–413. [see 3, 22 p.]

Nguyen, V.; Kearney, D.; Gioiosa, G. 2008. An implementation of membrane computing using reconfigurable hardware, *Computing and Informatics* 27(3): 551–569. [see 22 p.]

Nguyen, V.; Kearney, D.; Gioiosa, G. 2009. A region-oriented hardware implementation for membrane computing applications, in *International Workshop on Membrane Computing*, Springer, 385–409. [see 22 p.]

NIDDK 2016. *Diabetes Tests & Diagnosis* [interactive] [November 2016]. Available online at: https://www.niddk.nih.gov/health-information/diabetes/overview/tests-diagnosis. [see 3 p.]

Olczuk, D.; Priefer, R. 2018. A history of continuous glucose monitors (CGMs) in self-monitoring of diabetes mellitus, *Diabetes & Metabolic Syndrome: Clinical Research & Reviews* 12(2): 181–187. [see 83 p.]

Paun, A.; Paun, G. 2002. The power of communication: P systems with symport/antiport, *New Generation Computing* 20(3): 295–305. [see 10, 11, 12 p.]

Paun, A.; Perez-Jimenez, M. J.; Romero-Campero, F. J. 2006. Modeling signal transduction using P systems, in *Membrane Computing*, vol. 4361: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 100–122. [see 12 p.]

Paun, A.; Popa, B. 2006. P systems with proteins on membranes, *Fundamenta Informaticae* 72(4): 467–483. [see 12 p.]

Paun, G. 1997. DNA computing: Distributed splicing systems, in *Structures in Logic and Computer Science*, vol. 1261: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 353–370. [see 11 p.]

Paun, G. 1999. P systems with active membranes: Attacking NP complete problems, *Journal of Automata, Languages and Combinatorics* 6: 75–90. [see 10 p.]

Paun, G. 2000. Computing with membranes, *Journal of Computer and System Sciences* 108–143. [see 8, 9, 10, 11 p.]

Paun, G. 2001. Computing with membranes: Attacking NP-Complete problems, in *Unconventional Models of Computation*, Springer London, 94–115. [see 10 p.]

Paun, G. 2010. A quick introduction to membrane computing, *Journal of Logic and Algebraic Programming* 79(6): 291–294. [see 10, 11 p.]

Paun, G.; Rozenberg, G. 2002. A guide to membrane computing, *Theoretical Computer Science* 287(1): 73–100. [see 9, 10 p.]

Paun, G.; Rozenberg, G.; Salomaa, A. 2010. *The Oxford handbook of membrane computing*. Oxford University Press. [see 2, 11 p.]

Pescini, D.; Besozzi, D.; Mauri, G.; Zandron, C. 2006. Dynamical probabilistic P systems, *International Journal of Foundations of Computer Science* 17(01): 183–204. [see 12 p.]

Research and Markets 2018. *Global Smart Healthcare Market 2018–2022* [interactive] [July 2018]. Available online at: https://www.researchandmarkets.com/research/j6zw85/the_global_smart. [see 3 p.]

Romero-Aragon, J. C.; Sanchez, E. N.; Alanis, A. Y. 2014. Glucose level regulation for diabetes mellitus type 1 patients using FPGA neural inverse optimal control, in *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 1–7. [see 29 p.]

Romero-Campero, F. J.; Perez-Jimenez, M. J. 2008a. A model of the quorum sensing system in vibrio fischeri using P systems, *Artificial Life* 14(1): 95–109. [see 12 p.]

Romero-Campero, F. J.; Perez-Jimenez, M. J. 2008b. Modelling gene expression control using P systems: The lac operon, a case study, *Biosystems* 91(3): 438–457. [see 12 p.]

Ronak, B.; Fahmy, S. 2014. Efficient mapping of mathematical expressions into DSP blocks, in *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on*, 1–4. [see 55 p.]

Senthilkumar, B.; Umamaheswari, G. 2012. Improved noise removal algorithm implementation in FPGA for the breast cancer detection, in *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 1–2. [see 24 p.]

Siddique, N.; Adeli, H. 2015. Nature inspired computing: An overview and some future directions, *Cognitive Computation* 7(6): 706–714. [see 1 p.]

Sledevič, T.; Navakauskas, D. 2015. Towards optimal FPGA implementation of lattice-ladder neuron and its training circuit, in *Information, Electronic and Electrical Engineering (AIEEE), 2015 IEEE 3rd Workshop on Advances in*, 1–4. [see 54 p.]

Stašionis, L.; Serackis, A. 2013. A new approach for spectrum sensing in wideband, in *EUROCON, 2013 IEEE*, 125–132. [see 24 p.]

Stašionis, L.; Serackis, A. 2016. Experimental study of the spectrum sensor architecture based on discrete wavelet transform and feed-forward neural network, *Proceedings of the Romanian Academy, Series A* 17(2): 178–185. [see 24 p.]

Sun, X.; Alimohammad, A.; Trouborst, P. 2002. Modeling of FPGA local/global interconnect resources and derivation of minimal test configurations, in *Defect and Fault Tolerance in VLSI Systems, 17th IEEE International Symposium on*, 284–292. [see 24 p.]

Suzuki, Y.; Tanaka, H. 1997. Chemical oscillation in symbolic chemical systems and its behavioral pattern, in *Proceeding of International Conference on Complex Systems, New England Complex Systems Institute*, 1–7. [see 13, 14 p.]

Tolic, I. M.; Mosekilde, E.; Sturis, J. 2000. Modeling the insulin–glucose feedback system: The significance of pulsatile insulin secretion, *Journal of Theoretical Biology* 207(3): 361–375. [see 28 p.]

Trimberger, S. M. 2018. Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology, *IEEE Solid-State Circuits Magazine* 10(2): 16–29. [see 1, 3 p.]

Vassiliadis, V.; Dounias, G. 2009. Nature-inspired intelligence: a review of selected methods and applications., *International Journal on Artificial Intelligence Tools* 18: 487–516. [see 8 p.]

Vavouras, M.; Duarte, R. P.; Armato, A.; Bouganis, C.-S. 2016. A hybrid ASIC/FPGA fault-tolerant artificial pancreas, in *Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS), 2016 International Conference on*, 261–267. [see 29 p.]

Vouzis, P. D.; Bleris, L. G.; Arnold, M. G.; Kothare, M. V. 2009. A system-on-a-chip implementation for embedded real-time model predictive control, *IEEE Transactions on Control Systems Technology* 17(5): 1006–1017. [see 29 p.]

Whitmore, A.; Agarwal, A.; Da Xu, L. 2015. The internet of things–a survey of topics and trends, *Information Systems Frontiers* 17(2): 261–274. [see 3 p.]

Willmott, C. J.; Matsuura, K. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate research* 30(1): 79–82. [see 24 p.]

Willmott, C. J.; Matsuura, K.; Robeson, S. M. 2009. Ambiguities inherent in sums-of-squares-based error statistics, *Atmospheric Environment* 43(3): 749–752. [see 24 p.]

World Health Organization 2018. *The Top 10 Causes Of Death* [interactive] [24 May 2018]. Available online at: http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. [see 2 p.]

Xilinx 2014. *7 Series FPGA DSP48E1 Slice, User Guide* [Interactive] [November]. [see 53, 54 p.]

Xilinx 2018. *All Programmable 7 Series Product Selection Guide* [interactive] [1 September 2018]. Available online at: https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf. [see 25 p.]

Zandron, C.; Ferretti, C.; Mauri, G. 2001. Solving NP-complete problems using P systems with active membranes, in *Unconventional Models of Computation*, Springer, 289–301. [see 10 p.]

Zhabotinsky, A. M. 1991. A history of chemical oscillations and waves, *Chaos* 1(4): 379–386. [see 13 p.]

List of Scientific Publications by the Author on the Topic of the Dissertation

Papers in the Reviewed Scientific Journals

Kulakovskis, D. 2015. Application Prospects of Metabolic P System, *Mokslas – Lietuvos ateitis* 7(3): 285–290. ISSN 2029-2341. DOI: 10.3846/mla.2015.784.

Kulakovskis, D.; Navakauskas, D. 2016. Automated Metabolic P System Placement in FPGA, *Electrical, Control and Communication Engineering* 10(1): 5–12. ISSN 2255-9159. DOI: 10.1515/ecce-2016-0001.

Kulakovskis, D. 2019. Intraveninio gliukozės tolerancijos testo metabolinės P sistemos įgyvendinimas apibendrintuoju kombinaciniu būdu, *Mokslas – Lietuvos ateitis* 11: 1–5. ISSN 2029-2341. DOI: 10.3846/mla.2019.9429.

Papers in Other Editions

Kulakovskis, D.; Sledevič, T.; Gedminas, A.; Navakauskas, D. 2016. Alternative Implementations of Metabolic P System in FPGA, in *Proceedings of IEEE 4th Workshop on advances in information, electronic and electrical engineering*, 1–5. DOI: 10.1109/AIEEE.2016.7821816.

Kulakovskis, D.; Navakauskas, D. 2015. Automation of Metabolic P System Implementation in FPGA: A Case Study, in *Proceedings of IEEE 3rd Workshop on advances in information, electronic and electrical engineering*, 1–4. DOI: 10.1109/aieee.2015.7367289.

Summary in Lithuanian

Įvadas

Problemos formulavimas

Sparčiai vystantis elektronikos mokslo sričiai, daug dėmesio skiriama lygiagrečiųjų skaičiavimų įgyvendinimui. Tai patvirtina reikšmingai išaugusios kompanijų, tokių kaip Intel, investicijos į šią sritį. Viena iš aparatinio lygiagrečių skaičiavimų įgyvendinimo krypčių – lauku programuojamos loginės matricos (LPLM). Šių lustų gamybos procesai jau kurį laiką buvo sėkmingai vystomi kartu su kitomis lustų gamybos technologijomis. Taigi šiuo metu LPLM pasižymi dideliais skaičiavimo pajėgumais ir sąlyginai prieinama kaina

Spartus elektronikos vystymasis sukuria poreikį vertinti naujų taikymų ir įgyvendinimo būdų kokybę. Šiuo metu yra daug atskirų kokybės kriterijų naudojamų įgyvendinimo LPLM vertinimui, tačiau reikalingas naujas holistinis požiūris į kokybės vertinimą ir optimizavimą, atsižvelgiant į individualaus įgyvendinimo specifiką.

Kartu su elektronikos sritimi, sparčiai vystomos biologijos ir gamtinių skaičiavimų mokslo sritys. Tai skatina naujų intelektualiųjų elektroninių sistemų kūrimą. Gamtinių skaičiavimų tyrimų pavyzdžiais galima laikyti tokias sritis kaip konvoliuciniai neuronų tinklai ir membraniniai skaičiavimai. Remiantis gamtiniais skaičiavimais vykdomi Europinis *Human Brain* projektas ir JAV *SyNAPSE* programa.

Nauja infobiotikos mokslo sritis taip pat yra paremta gamtiniais skaičiavimais bei tokiomis sritimis kaip dirbtinė gyvybė ir skaitinės sintezės biologija. Čia apsijungia ir atranda sinergiją tokie žmonijos lūkesčiai, kaip noras: sukurti naujos kartos kompiuterius, veikiančius remiantis gamtoje egzistuojančiais dėsniais ar elgsena; *in silico* imituoti ir tirti gamtą bei jos reiškinius; panaudoti gamtinius išteklius ir gamtoje vykstančius procesus

skaičiavimams atlikti. P sistema gamtiniuose skaičiavimuose užima svarbią vietą. Viena iš jos nepriklausomai vystomų mokslo šakų – metabolinė P sistema (MP) – tai infobiotikos teorija ir diskrečiąja matematika grįstas metabolinio proceso aprašas.

MP sistemos jau buvo pritaikytos įvairiose gamtiniais skaičiavimais grįstose srityse. Vienas iš MP sistemų taikymų – intraveninis gliukozės tolerancijos testas (IVGTT) – modeliuoja gliukozės ir insulino sąveiką. Šios MP sistemos buvo kurtos panaudojant seniau sukurtus ir gerai ištirtus programinės įrangos įrankius.

Problema iškyla norint MP sistemas įgyvendinti ne programinėje, o aparatinėje įrangoje. Pirminė teorinė analizė ir MP sistemų tinkamumo įgyvendinimui aparatinėje įrangoje įrodymas jau yra pateikti, tačiau nėra tinkamai ištirtų metodų šiam įgyvendinimo uždaviniui spręsti, ypač daugelio lygiagrečių MP sistemų atveju. Šių sistemų LPLM įgyvendinimas suteiktų galimybę, susiejant elektronikos ir MP sistemų teorijos žinias ir pasitelkus naują požiūrį, našiau spręsti įvairias taikomąsias problemas. Siekiant šio tikslo būtina sukurti naujus daugelio MP sistemų transformavimo į LPLM struktūrinius elementus metodus.

Pagrindinė šiame darbe sprendžiama problema – žinių apie metabolinės P sistemos LPLM įgyvendinimo kokybę ir našių MP daugiasisteminių įgyvendinimo metodų trūkumas.

Šiai problemai išspręsti yra iškeliama ir įrodoma hipotezė: siekiant našaus MP daugiasisteminio veikimo realiuoju laiku, visos žinomos IVGTT MP sistemos gali būti apibendrintos viename LPLM įgyvendinime

Darbo aktualumas

Visuomenės poreikių požiūriu, diabeto gydymas yra viena svarbiausių sveikatos apsaugos sistemos problemų. Pasaulio sveikatos organizacijos duomenimis, diabetas yra viena pagrindinių mirties priežasčių pasaulyje. 2014 metais 422 milijonai žmonių visame pasaulyje sirgo diabetu. Šis skaičius išaugo nuo 108 milijonų 1980 metais. Tai rodo, kad žmonių, sergančių diabetu, skaičius sparčiai auga. Taip pat svarbu paminėti, kad tarp vyresnių nei aštuoniolikos metų žmonių sergamumas diabetu padidėjo nuo 4.7 % 1980 metais iki 8.6 % 2014 metais. Paėmus JAV kaip pavyzdį, pastebima, kad 29 milijonai žmonių šioje valstybėje (9.3 % visų gyventojų) serga diabetu. Tik 21 milijonui iš jų diabeto susirgimas yra diagnozuotas. Tai reiškia, kad bent 8 milijonams žmonių, arba 27 % sergančiųjų, šis susirgimas yra nediagnozuotas.

Diabeto diagnozės nustatymas bei gliukozės lygio kraujyje stebėjimas ir reguliavimas yra labai svarbūs milijonams žmonių sergančių šia liga arba turinčių kitų su gliukozės kiekiu kraujyje susijusių sutrikimų. Vieni iš dažniausiai naudojamų metodų diabeto diagnozavimui ir stebėjimui yra gliukozės tolerancijos testai, vienas iš kurių yra IVGTT. Taip pat daug dėmesio yra skiriama dirbtinės kasos sistemos kūrimui, kuri suteiktų galimybę automatiškai stebėti gliukozės kiekį kraujyje ir valdyti vaistų skyrimą diabetu sergantiems pacientams. Nors pastangos sukurti dirbtinę kasą trunka jau daugiau nei 50 metų, vis dar nėra patikimo ir tinkamo naudoti medicinoje prietaiso.

Atsižvelgiant į pažangą elektronikos srityje, dabar yra tinkamas metas giliau tyrinėti gliukozės stebėjimo įrenginius. Šiuo metu elektroninių įrenginių pasiūla sparčiai auga tokiose rinkose kaip daiktų internetas ar išmanus sveikatingumas. Ypač svarbu, kad šie įrenginiai vis dažniau yra naudojami medicinoje vietoje tradicinių gydymo metodų.

Sensoriais paremti elektroniniai įrenginiai vadinami dirbtinėmis kasomis tampa vis populiaresni diabeto gydyme. Šie įrenginiai buvo pademonstruoti kaip saugus ir efektyvus gliukozės stebėjimo ir insulino reguliavimo būdas, tačiau šioje srityje vis dar yra daug galimynių tobulėjimui.

Pastaruoju metu, LPLM tampa ekonomiškai pagrįstu įrankiu, skirtu naudoti nešiojamuose įrenginiuose. LPLM kaina mažėja, o jų integracijos laipsnis didėja tokiu tempu, kuris leido LPLM lustų talpai padidėti 10000 kartų nuo jų atsiradimo.

Elektroninių medicininių įrenginių prieinamumas lemia naujų socialinių reiškinių, tokių kaip savidiagnostikos, atsiradimą. Šio reiškinio populiarumas gali turėti reikšmingų pasekmių, nes tai gali pagerinti daugelio žmonių gyvenimo kokybę, ypač tokiose pasaulio vietose, kur profesionalios medicinos prieinamumas yra ribotas.

Žiūrint iš praktinės pusės, gliukozės ir insulino sąveikos modelių sudėtingumas tampa problema norint juos įgyvendinti įterptinėse sistemose. MP sistemos nenaudoja diferencialinių lygčių kaip kiti modeliai, todėl gali būti pritaikytos našiam realiame laike veikiančiam aparatiniam įgyvendinimui.

Šiuo metu trūksta MP sistemų aparatinio įgyvendinimo metodų, nors yra metodų skirtų panašioms membraninių skaičiavimų sistemoms. Teorinės įgyvendinimo galimybės jau yra išnagrinėtos, tačiau yra daugelio sistemų įgyvendinimo metodų bei jų kokybės charakteristikų tyrimų trūkumas. Šiuo metu egzistuojantys bendri LPLM įgyvendinimo kokybės kriterijai nėra specifiškai pritaikyti MP sistemoms ir turi būti peržiūrėti.

Tyrimų objektas

Tyrimų objektas yra specializuota realiu laiku veikianti metabolinė P sistema įgyvendinta lauku programuojamoje loginėje matricoje. Disertacijoje tiriami šie su tyrimo objektu susiję dalykai: įgyvendinimo kokybė ir įgyvendinimo būdai.

Darbo tikslas

Tikslas – pasiūlyti originalius lauku programuojamų loginių matricų technologija grįstus metabolinės P sistemos sprendimus, sukuriant ir ištiriant realaus laiko metabolinių procesų imitavimo ir testavimo elektroninę sistemą.

Darbo uždaviniai

Norint išspręsti iškeltą problemą ir įgyvendinti darbo tikslus, suformuluoti šie uždaviniai:

1. Remiantis metabolinės P sistemos teoriniais rezultatais ir pasaulinės praktikos įžvalgomis, pasiūlyti originalią metodiką metabolinės P sistemos konceptams transformuoti į lauku programuojamos loginės matricos struktūrinius elementus ir signalizacijos schemas.

2. Atskleisti šios transformacijos kokybines charakteristikas greitaveikos, sudėtingumo ir energijos sąnaudų kriterijų pagrindu.

3. Sukurti realaus laiko metabolinių procesų imitavimo ir testavimo elektroninę sistemą ir eksperimentiškai ją ištirti.

Tyrimų metodika

Dabartinė tyrinėjamų mokslo sričių būsena yra nagrinėjama atliekant literatūros ir technologijos analitinę apžvalgą. Infobiotikos teorija yra taikoma matematiniam metabolinio proceso aprašui. Aparatinio įgyvendinimo ir automatizavimo procesų tyrimui naudojama kokybės koncepcija ir metrikos. Tikslumo tyrimui pasitelkiamos vidutinės kvadratinės paklaidos (angl. *Root Mean Square Error – RMSE*) ir vidutinės absoliučiosios paklaidos (angl. *Mean Absolute Error – MAE*) kokybės metrikos. Kitoms metrikoms gauti naudojama LPLM aparatinių parametrų analizė. Daugiamačių kokybės charakteristikų vizualizacija naudojama skirtingų įgyvendinimų palyginimui. Kombinatorikos teorija bei instrukcijų planavimo, lygiagretinimo, fiksuoto kablelio skaičiavimų ir duomenų srautų grafikų technologijos yra taikomos kuriant LPLM įgyvendinimo metodus. Automatizuojant įgyvendinimus taip pat pasitelkiamas dvejetainės paieškos algoritmas ir duomenų struktūros formalizavimas. Kompiuterinė simuliacija yra naudojama kaip atskaitos taškas vertinant laboratorinių tyrimų ir matavimų procesus.

Remiantis ankstesnių tyrimų duomenimis, surinkti eksperimentams skirti intraveninio gliukozės tolerancijos testo duomenų rinkiniai. Simuliacijos atliekamos naudojant Matlab R2015b programinės įrangos paketą. Išmaniosios elektroninės sistemos įgyvendinamos naudojant Virtex-4 ir Zynq-7000 LPLM šeimas. Jų kūrimui ir eksperimentiniam tyrimui naudojamas Xilinx ISE Design Suite 14.7 programinės įrangos paketas kartu su originaliais sukurtais programiniais įrankiais.

Darbo mokslinis naujumas

1. Sukurtas pirmasis apibendrinto našaus kelių panašių metabolinių sistemų LPLM įgyvendinimo būdas, kuris leidžia LPLM dideliu mastu atlikti intraveninio gliukozės tolerancijos testo skaičiavimus.

2. Sudaryti metabolinių P sistemų įgyvendinimo LPLM kokybės kriterijai ir nauja jungtinė vertė, leidžianti įvertinti bendrą sistemos kokybę ir skirtingų metabolinės P sistemos įgyvendinimų tinkamumą.

3. Sukurtas naujas automatizuoto MP sistemos įgyvendinimo būdas, kuris įvertina fiksuoto kablelio aritmetikos skaičiavimų tikslumą ir suteikia galimybę greičiau nei prieš tai įgyvendinti LPLM skirtingas metabolines P sistemas ir paspartina matematinių modelių, tokių kaip IVGTT, tyrimus.

Darbo rezultatų praktinė reikšmė

Sukurtas MP sistema paremto IVGTT modelio įgyvendinimas naudojant pasiūlytą įgyvendinimo būdą. MP sistemos įgyvendinimo kokybės kriterijai leidžia pasirinkti reikiamą žodžio ilgį ir tinkamą LPLM lustą.

Įgyvendinta IVGTT sistema leidžia viename LPLM luste imituoti daugelio pacientų gliukozės ir insulino sąveikas. Duomenų rinkiniai gali būti keičiami ir skaičiavimai vykdomi iš naujo pakeičiant įvesties parametrus, taip suteikiant galimybę atlikti sparčius gliukozės ir insulino sąveikos tyrimus remiantis MP sistemomis.

Ginamieji teiginiai

1. Taikant pasiūlytą metabolinių P sistemų įgyvendinimo metodiką, visas šešias žinomas ir disertacijoje nagrinėtas intraveninio gliukozės tolerancijos testo metabolines P sistemas galima apibendrinti viename LPLM įgyvendinime su ne didesne nei 15% vidutine kvadratine paklaida naudojant ne trumpesnį nei 32 bitų žodžio ilgį.
2. Naujas metabolinių P sistemų jungtinės kokybės matas ir jo grafinis atvaizdas aiškiai atskiria įvairių metabolinių P sistemų lauku programuojamų loginių matricų įgyvendinimo kokybę ir sudaro sąlygas atskirai analizuoti šioms sistemoms būdingas kokybės charakteristikas.
3. Pasiūlytas apibendrintas kombinacinis intraveninio gliukozės tolerancijos testo metabolinių P sistemų įgyvendinimo būdas užtikrina nuo 2 iki 3 kartų didesnę spartą lyginant su apibendrintu srautinio įgyvendinimo būdu.

Disertacijos struktūra

Darbo apimtis yra 118 puslapiai, kuriuose yra pateikta: 63 formulė, 24 paveikslai, 11 lentelės, 2 algoritmai ir 7 pavyzdžiai. Disertacijoje remtasi 115 kitų autorių literatūros šaltiniais.

1. Metabolinės P sistemos modeliavimo ir įgyvendinimo apžvalga

Gamtiniai skaičiavimai yra plati kompiuterių mokslo sritis. Membraniniai skaičiavimai yra atskira šios srities kategorija, kuri kaip skaičiavimų objektą naudoja ląstelių membranas. Ši mokslo sritis pasitelkiama spręsti įvairias sudėtingas matematines problemas. Yra sukurta daug skirtingų membraninių skaičiavimų teorijos atmainų, iš kurių viena yra metabolinė P sistema.

Metabolinė P sistema buvo sukurta kaip specifinė P sistemos atmaina, kurioje dėmesys sutelkiamas į visose gyvose ląstelėse vykstantį metabolinį procesą. Šis procesas apibūdina ląstelės medžiagų apykaitą, t.y. kokios medžiagos patenka į ląstelę per jos membraną ir kokios medžiagos yra pašalinamos. Tokiu atveju membrana yra tarsi riba, kurią galima panaudoti įvairių procesų ir skaičiavimų modeliavimui.

MP sistema su reguliatorių rinkiniu gali būti aprašoma tokiu konstruktu:

$$M = (X, R, V, Q, \Phi, \nu, \mu, \tau, q_0, \delta)\,, \tag{S1.1}$$

čia X yra medžiagų aibė, R yra reakcijų aibė, V yra parametrų aibė, Q yra būsenų aibė, Φ yra regulatorių aibė, ν yra sveikas skaičius nurodantis molekulių kiekį, μ yra masę priskirianti funkcija, τ yra laiko intervalas, q_0 yra pradinė būsena ir δ yra sistemos dinamika. MP sistema be būsenų aibės Q ir dinamikos, yra MP grafas, kuris gali būti atvaizduotas grafiškai. Papildomai praleidus τ, ν ir μ, gaunama MP gramatika.

Metabolinės P sistemos dažniausiai naudojamos realių periodinių funkcijų aproksimacijai. Kiekvienai aproksimuojamai funkcijai reikia sudaryti atskirą MP sistemą. Šis MP

sistemos taikymas yra naudojamas įvairiose srityse, tokiose kaip matematika, biologija, chemija ir kitos. Viena iš biologijoje naudojamų MP sistemų aprašo gliukozės ir insulino sąveiką intraveninio gliukozės tolerancijos testo metu.

Egzistuoja keli MP modeliai skirti IVGTT procedūrai. Skirtingo tipo IVGTT MP sistemos gali skirtis aritmetinių operacijų skaičiumi ir koeficientų reikšmėmis. Paprasčiausia iš jų yra ši MP sistema:

$$\begin{aligned} r_1 &: \emptyset \to G, \quad \phi_1 = 0,6\,; \\ r_2 &: G \to \emptyset, \quad \phi_2 = 0,12G + 1,6 \cdot 10^{-6} G^2 I\,; \\ r_3 &: \emptyset \to I, \quad \phi_3 = 49,9 + 0,1G^3\,; \\ r_4 &: I \to \emptyset, \quad \phi_4 = 0,84I\,. \end{aligned} \tag{S1.2}$$

IVGTT MP sistemoje yra dviejų tipų parametrai: taisyklės (r) ir reguliatoriai (ϕ). Taisyklės aprašo medžiagų (gliukozės G ir insulino I) sąveikas sistemoje. Medžiagų kiekis kinta priklausomai nuo to, kurios taisyklės yra taikomos. Reguliatoriai nurodo tikslų šio pokyčio dydį. IVGTT sistema yra viena iš svarbiausių ir daugiausiai žadančių MP sistemų taikymo galimybių. Įgyvendinus šią sistemą elektroniniame matavimo įrenginyje atsirastų galimybė pagelbėti diabeto tyrimams ir gydymui.

Nors MP sistemos gali būti taikomos daugelyje sričių, tam reikia jas įgyvendinti kokiame nors įrenginyje, gebančiame atlikti kompiuterinius skaičiavimus. Iki šiol visi MP sistemos įgyvendinimo būdai buvo vykdomi naudojant programinę įrangą, kuri gali būti pritaikoma naudojant sąlyginai galingus bendrosios paskirties kompiuterius. Vis dėlto šis sprendimas ne visuomet yra tinkamas, ypač tada kai MP skaičiavimus reikia atlikti mažesniame, labiau specializuotame įrenginyje. Įgyvendinus MP sistemas įterptiniame įrenginyje būtų sudaryta galimybė atlikti realaus laiko MP sistemų skaičiavimus nešiojamame prietaise.

Metabolinės P sistemos gali būti programiškai įgyvendintos naudojant specializuotą programinę įrangą arba sąlyginai paprastus formulių skaičiavimus atliekant MATLAB™ terpėje. Iki šiol nėra žinomų MP sistemos įgyvendinimų aparatinėje įrangoje, nors tokie įgyvendinimai buvo atlikti naudojant P sistemas.

Norint efektyviai įgyvendinti MP sistemas aparatinėje įrangoje, reikalingas šio įgyvendinimo kokybės vertinimas. Kokybė gali būti vertinama skirtingais būdais ir naudojant tam skirtą kriterijų rinkinį. Vertinimo kriterijai turi būti pritaikyti aparatiniam MP sistemos įgyvendinimui, nes jis turi savų iššūkių lyginant su programiniu įgyvendinimu.

Tam, kad teisingai įvertinti įgyvendinimo kokybę, turi būti atsižvelgta į konkretaus įgyvendinimo tikslą. MP sistemos atveju šis tikslas yra teisingas ir našus aritmetinių operacijų taikymas paremtas MP sistemos formulėmis. Tai galima suskaidyti į tris atskiras grupes: skaičiavimų tikslumas, skaičiavimų sparta ir aparatinės įrangos energijos suvartojimas. Metabolinės P sistemos įgyvendinimo LPLM kokybės charakteristikos turi būti atskleistos.

Kokybės vertinimo kriterijų tinkamumui patvirtinti turi būti sukurta ir ištirta realaus laiko metabolinių procesų imitavimo ir testavimo sistema. Šiam tikslui pasiekti tinkamas įrankis yra lauku programuojamos loginės matricos.

2. Metabolinės P sistemos aparatinio įgyvendinimo kokybės kriterijai

Prieš įgyvendinant metabolines P sistemas aparatinėje įrangoje, būtina nustatyti kriterijus, kuriais šis įgyvendinimas bus vertinamas. Vertinimo kriterijai turi tinkamai apibūdinti įgyvendinimo kokybę. Bendros įgyvendinimo kokybės apibūdinimui nepakanka vieno matavimo ar parametro. Pilnai apibūdinti kokybę gali tik skirtingų kokybės aspektų kombinacija. Tas pats įgyvendinimas gali pasižymėti tam tikrais pranašumais pagal vienas metrikas, tačiau taip pat gali pasižymėti ir trūkumais pagal kitas metrikas. Todėl reikia atsižvelgti į kokybės metrikų visumą.

Dauguma išnagrinėtų aparatinių įgyvendinimų yra vertinami pagal LPLM išteklių panaudojimą, skaičiavimo spartą, energijos suvartojimą, patikimumą ir kitus parametrus. Nors daug skirtingų kokybės kriterijų yra naudojami išnagrinėtų gliukozės ir insulino stebėjimo sistemų įgyvendinimų LPLM vertinimui, nei vienas iš šių kriterijų nėra specifiškai pritaikytas MP sistemoms. Taip pat nėra vienareikšmiško sutarimo, kurie kriterijai yra svarbiausi. Dėl to reikia sudaryti kokybės vertinimo kriterijų rinkinį specifiškai pritaikytą MP sistemų įgyvendinimui LPLM. Šie kriterijai vėliau bus naudojami, norint įvertinti MP sistemų įgyvendinimą skirtingais būdais.

MP sistemų įgyvendinimui LPLM pritaikyti kokybės vertinimo kriterijai buvo atrinkti išnagrinėjus kitus aparatinių įgyvendinimų pavyzdžius ir atsižvelgus į MP sistemos ypatumus. Iš viso buvo atrinkti penki kriterijai, kurie leidžia apibūdinti bendrą MP sistemos įgyvendinimo LPLM kokybę. Bendra įgyvendinimo kokybė užrašoma kaip penkių normuotų (rėžiuose nuo 0 iki 100) kokybės kriterijų aibė:

$$Q_{\mathrm{MP}} = \{Q_{\mathrm{A}}, Q_{\mathrm{T}}, Q_{\mathrm{R}}, Q_{\mathrm{P}}, Q_{\mathrm{I}}\} \,, \tag{S2.1}$$

čia Q_{MP} yra bendra MP sistemos įgyvendinimo LPLM kokybė; Q_{A} yra skaičiavimo tikslumo kokybė; Q_{T} yra rezultatų generavimo spartos kokybė; Q_{R} yra LPLM išteklių panaudojimo kokybė; Q_{P} yra LPLM energijos suvartojimo kokybė; Q_{I} yra yra LPLM sąsajos naudojamos įgyvendinimui sudėtingumas.

Tikslumo kokybė nusako atvirkštinę skaičiavimo paklaidai reikšmę, kai rezultatai lyginami su pasirinktu atskaitos tašku. Klaidos maksimali vertė priklauso nuo MP sistemos medžiagų maksimalių verčių, todėl gali skirtis priklausomai nuo pasirinktos MP sistemos tipo. Momentinės vertės taip pat gali skirtis ir tos pačios MP sistemos viduje, tarp skirtingų medžiagų. Dėl to, norint korektiškai įvertinti ir palyginti skirtingų MP sistemų ir skirtingų medžiagų skaičiavimo paklaidas, būtina naudoti normuotas paklaidų vertes. Sunormavus paklaidų vertes, gaunama tikslumo kokybės vertė kintanti nuo 0 iki 100, kaip ir kitos kokybės vertės.

Skaičiavimų paklaidos reikšmė gali būti apskaičiuojama naudojant skirtingus metodus. Apskaičiuotų medžiagų verčių išvesties signalas yra aprašomas vektoriumi, kai MP sistemoje iš viso yra viena medžiaga, arba matrica, kai MP sistemoje yra daugiau nei viena medžiaga. Dėl to MP sistemos įgyvendinimo tikslumo kokybė aprašoma:

$$Q_{\mathrm{A}} = (1 - E) \times 100 \,[\%] \,, \tag{S2.2}$$

čia E yra išreikšta kaip normuota vidutinė kvadratinė paklaida

$$E_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\widehat{x}(n)-x(n)\right)^2} \Big/ \left(x^{\top}-x^{\perp}\right), \tag{S2.3}$$

arba normuota vidutinė absoliučioji paklaida

$$E_{\text{MAE}} = \frac{1}{N}\sum_{n=1}^{N}\left|\widehat{x}(n)-x(n)\right| \Big/ \left(x_{\top}-x_{\perp}\right), \tag{S2.4}$$

čia $x(n)$ – etaloninė MP sistemos medžiagos reikšmė diskrečiu laiko momentu n; $\widehat{x}(n)$ – aproksimuota MP sistemos medžiagos reikšmė diskrečiu laiko momentu n; N – MP sistemos medžiagos aproksimacijos rezultatų reikšmių vektoriaus ilgis. Tokiu atveju, kai MP sistemoje yra daugiau nei viena medžiaga ir naudojama rezultatų matrica, paklaidų reikšmės turi būti papildomai suvidurkinamos per visas matricos eilutes.

Naudojami *RMSE* paklaidų vektoriaus ekstremumai yra mažiau reikšmingi bendrai paklaidos reikšmei lyginant su *MAE*. Daugeliu MP sistemų taikymų atveju ši savybė gali būti naudinga, tačiau tai daugiausiai priklauso nuo konkretaus taikymo ir pasirinktos įgyvendinti MP sistemos tipo. *MAE* gali būti naudinga tais atvejais, kai norima išryškinti vienos medžiagos skaičiavimo paklaidų didžiausias vertes.

MP sistemų skaičiavimo paklaida labiausiai priklauso nuo pasirinkto žodžio ilgio. Kadangi LPLM įgyvendinime yra naudojama fiksuoto kablelio aritmetika, žodžio ilgis susideda iš dviejų dalių – sveikosios ir trupmeninės. Jas reikia parinkti atsižvelgiant į įgyvendinamos MP sistemos skaitinių reikšmių amplitudę.

Skaičiavimo spartos kokybė atvaizduoja kaip arti maksimalaus galimo elektroninės sistemos taktinio dažnio yra apskaičiuotų MP sistemos medžiagų reikšmių generavimo sparta:

$$Q_{\text{T}} = \frac{f_{\text{val}}}{f^{\top}} \times 100\ [\%], \tag{S2.5}$$

čia f_{val} – MP sistemos medžiagų reikšmių generavimo dažnis (Hz); $f^{\top}$ – didžiausias įmanomas pasirinkto LPLM lusto reikšmių generavimo dažnis (Hz), kuris yra glaudžiai susijęs su taktiniu dažniu, negali būti viršytas ir yra nurodomas specifikacijose pateikiamose LPLM lustų gamintojų (dažniausiai svyruoja nuo 100 iki 300 MHz).

MP sistemos medžiagų reikšmių generavimo sparta f_{val} nurodo kaip dažnai yra gaunama nauja MP sistemos reikšmė. f_{val} išlieka nepakitęs net jei MP sistemoje yra kelios medžiagos. Taip yra dėl MP sistemų specifikos, kur norint paskaičiuoti kitą kurios nors medžiagos reikšmę, reikalingos tik ankstesnių skaičiavimo ciklų reikšmės. Tai reiškia, kad atsiranda galimybė pasinaudoti LPLM lustų privalumu, leidžiančiu lygiagrečiai skaičiuoti visų MP sistemos medžiagų reikšmes.

Išteklių panaudojimo kokybė nusako panaudotų LPLM elementų dalį. Trys pagrindiniai LPLM elementai, kurių panaudojimas yra vertinamas, yra šie: skaitmeninio signalų apdorojimo (SSA) elementai (angl. *Digital Signal Processor – DSP*), peržvalgos lentelės (PL) (angl. *Look-Up Table – LUT*), blokinė adresuojama atmintis (angl. *Block RAM – BRAM*). Tam, kad palyginti bendrą išteklių panaudojimą, visų šių elementų panaudojimo

skaičiai turi būti konvertuojami į ekvivalentinį dydį. Todėl yra naudojamas ekvivalenčių peržvalgos lentelių parametras:

$$Q_{\mathrm{R}} = \frac{N_{\mathrm{maxLUT}} - N_{\mathrm{eqLUT}}}{N_{\mathrm{maxLUT}}} \times 100\ [\%]\,, \tag{S2.6}$$

čia bendri panaudoti ištekliai išreikšti, ekvivalentiniais PL vienetais

$$N_{\mathrm{eqLUT}} \equiv w_{\mathrm{LUT}} N_{\mathrm{LUT}} + w_{\mathrm{DSP}} N_{\mathrm{DSP}} + w_{\mathrm{BRAM}} N_{\mathrm{BRAM}}\,, \tag{S2.7}$$

čia N_{LUT} – išnaudotų logikos operacijų išteklių skaičius; N_{DSP} – išnaudotų aritmetikos operacijų išteklių skaičius; N_{BRAM} – išnaudotų atminties išteklių skaičius; N_{maxLUT} – iš viso galimų panaudoti PL skaičius; w – svoriai (koeficientai), nurodantys ryšį su ekvivalenčiais PL ištekliais.

Energijos suvartojimo kokybė apibrėžia ryšį tarp galios, reikalingos MP sistemos LPLM įgyvendinimui, ir maksimalios galios, kurią sugeba suteikti LPLM lustas, atsižvelgiant į lusto šiluminį galios dizainą.

Sąsajos sudėtingumo kokybė nusako panaudotų LPLM lusto įvesties ir išvesties blokų dalį. Šie blokai yra naudojami pradiniams parametrams perduoti į lustą ir galutiniams skaičiavimų rezultatams perduoti į išorinę aplinką.

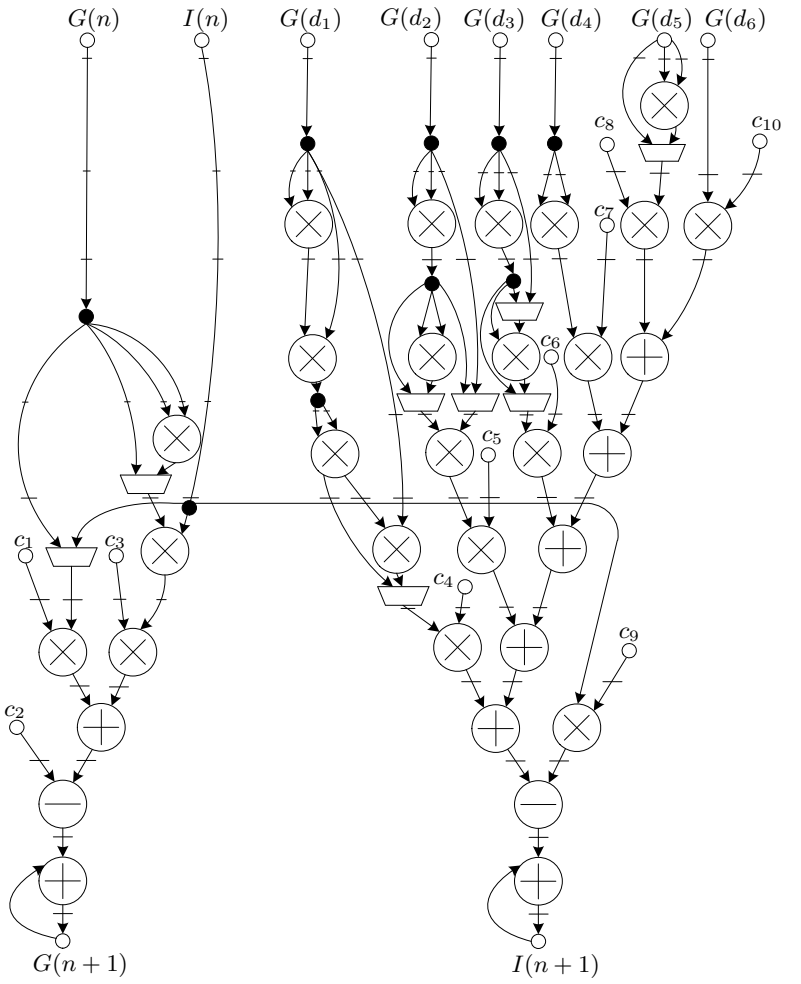
Pasiūlyti MP sistemos įgyvendinimo kokybės vertinimo kriterijai gali būti naudojami tais atvejais, kai skirtingos MP sistemos yra įgyvendinamos tokio pačio tipo LPLM. Kokybės kriterijų reikšmės turi būti transformuojamos į kitą atskaitos tašką, kai norima palyginti įgyvendinimus skirtingo tipo LPLM lustuose.

3. Metabolinės P sistemos aparatinio įgyvendinimo būdai

Metabolinių P sistemų įgyvendinimui LPLM gali būti pasitelkiami keli skirtingi metodai, kurie pasižymi vienokiais ar kitokiais trūkumais, priklausančiais nuo konkretaus taikymo. MP sistemų specifika taip pat turi įtakos pasirenkant konkretų įgyvendinimo metodą. Į tai būtina atsižvelgti vertinant metodo tinkamumą.

Nauja originalaus dizaino *JSON* formato duomenų struktūra buvo sukurta palengvinti MP sistemų užrašymą programinei logikai suprantamu būdu. *JSON* formatas buvo pasirinktas dėl nesudėtingo konvertavimo ir suderinamumo su daugeliu šiuolaikinių programavimo kalbų. Siūloma duomenų struktūra yra glaudžiai susijusi su MP sistemų struktūra atvaizduota MP grafo išraiška. Ši struktūra suteikia galimybę, panaudojus programinės įrangos įrankius, automatizuoti MP sistemų konvertavimą iš vieno formato arba kalbos į kitą.

Sukurtas automatizuoto įgyvendinimo būdas buvo pritaikytas MP sistemų *VHDL* kodo generavimo įrankyje, parašytame Python interpretuojama programavimo kalba. Šis įrankis geba konvertuoti MP sistemą, aprašyta *JSON* duomenų struktūra, į LPLM architektūrą išreikštą *VHDL* programavimo kalba. Gautas *VHDL* kodo fragmentas gali būti panaudojamas kaip atskiras komponentas arba įterpiamas į didesnes *VHDL* programas. Tai paspartina ir supaprastina MP sistemų įgyvendinimo LPLM procesą.



S3.1 pav. Intraveninio gliukozės tolerancijos testo metabolinė P sistema įgyvendinta apibendrintu srautiniu būdu

Kombinacinis įgyvendinimo būdas pasitelkia tradicinį elgesio procesą su sinchroniniu perkrovimu. Šis įgyvendinimo būdas panaudoja LPLM elementus, tokius kaip peržvalgos lentelės ir skaitmeninių signalų procesoriai, kad sukurtų nuoseklią loginių operacijų grandinę. Konkreti LPLM elementų konfigūracija nustatoma *VHDL* kodo optimizacijos ir sintezės fazėje, kurią atlieka LPLM lusto gamintojo pateiktas kodo sintezatorius.

Pagrindinis kombinacinio įgyvendinimo privalumas yra tai, kad vėlinimas visada yra lygus vieno takto trukmei. Tai reiškia, kad visos skaičiavimo operacijos yra atliekamos nuosekliai vieno takto metu. Kita vertus, tai taip pat lemia maksimalaus įgyvendinimo veikimo dažnio sumažėjimą, nes vieno takto metu yra atliekama palyginti daug operacijų. Tai ypač pastebima įgyvendinant sudėtingas MP sistemas, turinčias palyginti daug me-

džiagų. Į šią kombinacinio įgyvendinimo būdo savybę būtina atkreipti dėmesį taikant jį MP sistemoms.

Dauguma šiuolaikinių LPLM turi dedikuotus SSA elementus sparčių aritmetinių operacijų įgyvendinimui. Toks SSA blokas gali būti naudojamas duomenų apdorojimui iš skirtingų šaltinių bei gali būti dinamiškai perkonfigūruotas, įgyvendinant kitokias duomenų apdorojimo komandas. SSA elementai yra specialiai sukurti maksimaliam srautinio duomenų apdorojimo našumui pasiekti.

Vieno SSA elemento įgyvendinimo metodas panaudoja tik vieną LPLM skaitmeninių signalų procesorių visoms aritmetinėms operacijoms atlikti. Tokiu būdu yra užtikrinamas kuo mažesnis LPLM išteklių panaudojimas ir pasiekiamas didelis skaičiavimo greitis. Kita vertus, gali ženkliai padidėti vėlinimas, ypač kai įgyvendinama sąliginai sudėtinga MP sistema, nes visiems skaičiavimams atlikti reikia daugiau taktų. Taip pat dėl *DSP* architektūros yra ribojamas maksimalus žodžio ilgis, kurį galima panaudoti vienam aritmetinės operacijos nariui užrašyti.

Srautinis įgyvendinimo metodas pasižymi tuo, kad, skirtingai nuo kombinacinio įgyvendinimo metodo, neatlieka visų skaičiavimų vieno takto metu. Skaičiavimai yra lygiagretinami, vieno takto metu atliekant kuo įmanoma daugiau nepriklausomų operacijų. Šis įgyvendinimo metodas išnaudoja MP sistemos skaičiavimų lygiagretumo savybę. Daugeliu atveju, MP sistemos skaičiavimai gali būti atliekami keliais lygiagrečiais žingsniais, ypač kai MP sistema sudaryta iš kelių skirtingų medžiagų.

Aptarti bendri įgyvendinimo metodai yra tinkami vienos MP sistemos įgyvendinimui LPLM. Vis dėlto, dažniausiai naudoti LPLM vienos MP sistemos skaičiavimui yra neefektyvu. Egzistuoja taikymai, kur dėl didesnio patikimumo arba dėl naudojamų skirtingų tyrimo objektų yra būtinybė vienu metu skaičiuoti daugiau nei vieną MP sistemą. Tokiu atveju bendrų įgyvendinimo būdų LPLM išteklių panaudojimas tiesiškai priklauso nuo įgyvendinamų MP sistemų skaičiaus. Kelių įgyvendinimų apjungimo metodas gali būti panaudotas našumui padidinti, kai įgyvendinamos kelios tokio paties tipo MP sistemos atmainos.

Pagrindinis apibendrinto MP sistemų įgyvendinimo tikslas yra sudaryti bendrą MP sistemos išraišką, kuri aprėpia visas tam tikro MP sistemos tipo atmainas. Šios atmainos gali turėti skirtingus koeficientus arba skirtingas aritmetines operacijas reguliatorių išraiškose. Nesikeičiantys parametrai yra taisyklių ir medžiagų skaičiai. Tiek kombinacinio, tiek srautinio įgyvendinimo būdai gali būti pritaikyti apibendrintam MP sistemos įgyvendinimui.

Apibendrintam įgyvendinimui pasirinkta IVGTT sistema turi kelias skirtingas MP gramatikos atmainas. IVGTT MP sistemos yra sudaromos aproksimuojant skirtingų pacientų tyrimų duomenų rinkinius. Šešios skirtingos MP sistemos yra panaudotos apibendrintam įgyvendinimui, pavaizduotam S3.1 pav. Nors skirtingų IVGTT sistemų MP gramatika nėra identiška, pačių sistemų struktūra yra labai panaši. Skiriasi tik reguliatorių išraiškos ir koeficientų reikšmės, kurios priklauso nuo pasirinktų duomenų rinkinio. Šis dėsningumas išliktų nepakitęs sudarius IVGTT MP sistemą ir pagal naujus tyrimų duomenis. Šešios skirtingos reguliatorių išraiškos panaudotos šiame įgyvendinime pavaizduotos S3.1 lentelėje.

Skirtingos IVGTT MP sistemos atmainos pasižymi panašiais įvesties parametrais ir aritmetinių operacijų eiliškumu. Dėl šios priežasties, atskiros apibendrinto įgyvendinimo šakos gali būti įjungiamos, išjungiamos arba panaudojamos kelioms skirtingoms IVGTT

S3.1 lentelė. Intraveninio gliukozės tolerancijos testo metabolinių P sistemų reguliatoriai

MP sistema	Reguliatorius
IVGTT-1	$\phi_1 = 0.6$ $\phi_2 = 0.12G + 1.6 \cdot 10^{-6} G^2 I$ $\phi_3 = 49.9 + 0.1G^3$ $\phi_4 = 0.84I$
IVGTT-2	$\phi_1 = 0.6$ $\phi_2 = 0.12G + 1.6 \cdot 10^{-6} G^2 I$ $\phi_3 = 1.5 \cdot 10^{-5} G^6 + 0.25G^2_{-6} + 0.17G^2_{-8} + 2.65G_{-16} +$ $+ 3.6G_{-26}$ $\phi_4 = 0.65I$
IVGTT-3	$\phi_1 = 0.011$ $\phi_2 = 6.6 \cdot 10^{-5} GI$ $\phi_3 = 0.5G^2_{-4}$ $\phi_4 = 0.16I$
IVGTT-4	$\phi_1 = 0.056$ $\phi_2 = 5.2 \cdot 10^{-4} I + 8.1 \cdot 10^{-5} GI$ $\phi_3 = 3.76 \cdot 10^{-6} G^7 + 0.74G^2_{-8} + 0.02G^3_{-20} + 0.21G^2_{-40} +$ $+ 10^{-4} G^5_{-68}$ $\phi_4 = 0.49I$
IVGTT-5	$\phi_1 = 0.12$ $\phi_2 = 0.02G + 1.9 \cdot 10^{-4} GI$ $\phi_3 = 0.04G^3_{-2} + 3.3 \cdot 10^{-5} G^6_{-6} + 0.44G^2_{-20} + 0.04G^3_{-24}$ $\phi_4 = 0.5I$
IVGTT-6	$\phi_1 = 0.11$ $\phi_2 = 6.2 \cdot 10^{-4} GI$ $\phi_3 = 0.1G^2_{-2} + 0.9G_{-6} + 1.07G_{-10} + 2.4 \cdot 10^{-4} G^4_{-24} +$ $+ 5.4 \cdot 10^{-7} G^6_{-32} + 5.3 \cdot 10^{-8} G^7_{-34}$ $\phi_4 = 0.4I$

MP sistemoms. Skaičiuojama IVGTT MP sistema pasirenkama nustatant reikiamą įėjimo signalą. Tokiu būdu, vienoje LPLM įgyvendinant kelias IVGTT MP sistemas, apibendrintas įgyvendinimo būdas įgauna pranašumą lyginant su atskirų pavienių sistemų įgyvendinimo būdais.

Svarbus apibendrinto įgyvendinimo aspektas yra tai, kad visi įmanomi koeficientai turi būti aprašyti visiems sistemų tipams, net jei tos sistemos reguliatoriaus išraišką sudaro mažesnis operacijų skaičius. Esant nenaudojamoms operacijoms, jų koeficientams priskiriamos 0 vertės, taip praleidžiant šias operacijas. Patys koeficientai yra saugomi "sfixed" formatu, kuris nurodo, kad yra naudojama fiksuoto kablelio aritmetika su ženklo bitu.

Visas aptartas apibendrintos IVGTT MP sistemos veikimas yra pagrįstas bendrų LPLM išteklių dalinimusi skirtingoms sistemoms skaičiuoti. Norimas skaičiuoti sistemos tipas gali būti keičiamas kiekvieno takto metu. Tai reiškia, kad, įgyvendinus apibendrintą sistemą, atsiranda galimybė skaičiuoti visų tipų IVGTT MP sistemas išvengiant poreikio kiekvieną iš jų įgyvendinti atskirai.

4. Metabolinės P sistemos aparatinio įgyvendinimo tyrimas

Metabolinės P sistemos įgyvendinimo LPLM tyrimas atliekamas dviem etapais: įgyvendinant vieną MP sistemą ir įgyvendinant kelias MP sistemas vienu metu. Tyrimui pasirinkta IVGTT sistema, panaudojant sudarytą šešių IVGTT sistemos atmainų duomenų rinkinį. Įgyvendinant vieną MP sistemą yra naudojami bendri LPLM įgyvendinimo metodai. Tuo tarpu, įgyvendinant kelias MP sistemas vienu metu, naudojamas sukurtas apibendrinto įgyvendinimo būdas.

Skaičiavimo tikslumas priklauso nuo pasirinkto žodžio ilgio ir įgyvendinamos MP sistemos sudėtingumo. Aritmetinių operacijų eiliškumas buvo išlaikomas pastovus įgyvendinant IVGTT MP sistemas skirtingais metodais. Todėl gauti skaičiavimo tikslumo rezultatai gali būti taikomi nepriklausomai nuo to, kokiu metodu yra įgyvendinta MP sistema.

Lyginant gliukozės ir insulino išėjimo signalus, gautus įgyvendinus visas IVGTT sistemas LPLM, galima pastebėti, kad insulino signalas kinta staigiau. Tai reiškia, kad šias sistemas įgyvendinus LPLM naudojant fiksuotą žodžio ilgį, galima tikėtis didesnės paklaidos insulino vertėse. Taip pat galima pastebėti, kad IVGTT1 sistema pasižymi tolygiausiu išėjimo signalu, kas turėtų padidinti įgyvendinimo tikslumą.

Pasirinktos konkrečios IVGTT sistemos sudėtingumas daro įtaką galutinių rezultatų tikslumui. Laikoma, kad leistina maksimali normuota *RMSE* paklaida turi neviršyti $15\,\%$, o maksimali normuota *MAE* paklaida – $10\,\%$. Tam, kad pasiekti šį tikslą visoms įgyvendintoms IVGTT sistemoms, būtina naudoti bent 32 b žodžio ilgį, nors atskirai IVGTT1 ir IVGTT2 sistemoms pakanka 18 b, o IVGTT5 ir IVGTT6 sistemoms pakanka 24 b.

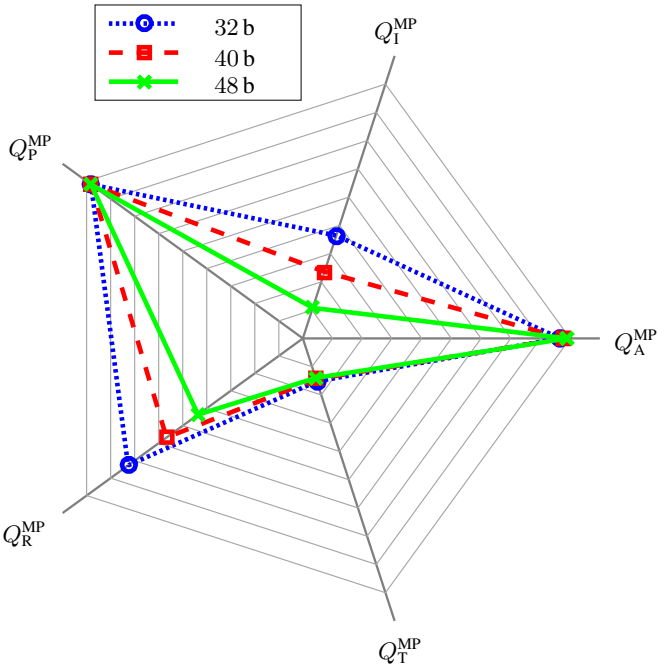
Dėl užsibrėžto tikslumo, kurį riboja žodžio ilgis, vieno SSA elemento įgyvendinimo metodas yra tinkamas tik IVGTT1 ir IVGTT2 sistemoms. Kitų IVGTT sistemos tipų tikslumas yra per mažas naudojant vieno SSA elemento metode taikomą 18 b žodžio ilgį.

Todėl, kad vieno SSA elemento metodas yra tinkamas įgyvendinti IVGTT sistemas tik ribotais atvejais (tik 2 iš 6 sistemų gali būti pakankamai tiksliai įgyvendintos), tolimesniuose tyrimuose yra naudojami tik kiti du aptarti metodai. 32 b žodžio ilgis yra pasirinktas tolimesniems tyrimams, nes tokio ilgio pakanka tiksliai įgyvendinti visas pasirinktas IVGTT sistemas.

Kombinacinis įgyvendinimas pasiekia didesnį visų įgyvendintų MP sistemų skaičiavimo greitį, remiantis paskaičiuota f_{val} reikšme. Nors šis įgyvendinimas pasiekia mažesnį maksimalų veikimo dažnį lyginant su srautiniu įgyvendinimu, šį trūkumą atsveria tai, kad rezultatų vėlinimas yra lygus tik vienam taktui.

Lyginant LPLM išteklių panaudojimo metrikas, pastebima, kad nei vienas iš tirtų metodų nėra našiausias visoms MP sistemoms. Vis dėlto, dažniau našesnis yra kombinacinis metodas, kuriuo įgyvendintos 4 iš 6 IVGTT sistemų panaudojo mažiau LPLM išteklių. Likusiais dviem atvejais, kai našesnis buvo srautinis įgyvendinimas, išteklių panaudojimas skyrėsi nežymiai, atitinkamai $2,1\,\%$ IVGTT1 sistemai ir $0,4\,\%$ IVGTT3 sistemai, lyginant su kombinaciniu įgyvendinimu.

Apibendrintas MP sistemų įgyvendinimo metodas suteikia galimybę įgyvendinti LPLM tik vieną komponentą, kuris geba paskaičiuoti skirtingo tipo MP sistemas. Apibendrintam IVGTT MP sistemų įgyvendinimui buvo pritaikyti kombinacinis ir srautinis įgyvendinimo metodai. Apibendrintų kombinacinio ir srautinio įgyvendinimų LPLM išteklių panaudiji-



S4.1 pav. Apibendrinto kombinacinio įgyvendinimo su skirtingais žodžio ilgiais kokybių palyginimas

mo kokybė skiriasi tik $0,02$ %. Abiejų apibendrintų įgyvendinimų panaudotų SSA elementų skaičius yra toks pats su visais žodžio ilgiais, o panaudotų PL skaičius kinta nežymiai. Tai skiriasi nuo vienos MP sistemos įgyvendinimo atvejų, kur kombinacinis įgyvendinimas pasižymi geresne išteklių panaudojimo kokybe.

Maksimalus veikimo dažnio skirtumas yra pakankamai ženklus lyginant abu apibendrintus įgyvendinimus. Žiūrint į 32 b žodžio ilgio įgyvendinimus, apibendrintas srautinis įgyvendinimas pasižymi 5 kartus didesniu maksimaliu dažniu. Vis dėlto, apibendrinto kombinacinio įgyvendinimo rezultatų generavimo sparta yra didesnė, nes vėlinimas visada lygus 1 taktui, kai tuo tarpu apibendrinto srautinio įgvendinimo vėlinimas siekia 10 taktų.

Paskaičiuotos kokybės reikšmės pavaizduotos S4.1 lentelėje. Tikslumas $Q_{\mathrm{A}}^{\mathrm{MP}}$ yra toks pats tiek apibendrinto srautinio, tiek apibendrinto kombinacinio įgyvendinimo atveju, nes abiem atvejais yra naudojamas toks pats 32 b žodžio ilgis.

Taip pat ir sąsajos sudėtingumo kokybė $Q_{\mathrm{I}}^{\mathrm{MP}}$ yra vienoda, nes išvesties signalai nesiskiria tarp įgyvendinimo metodų. Apibendrintas srautinis įgyvendinimas pasižymi nedaug didesne išteklių panaudojimo kokybe $Q_{\mathrm{R}}^{\mathrm{MP}}$, tačiau apibendrintas kombinacinis įgyvendinimas yra pranašesnis skaičiavimo greičio $Q_{\mathrm{T}}^{\mathrm{MP}}$ ir energijos suvartojimo $Q_{\mathrm{P}}^{\mathrm{MP}}$ kokybių atžvilgiu.

S4.1 lentelė. Bendros 32 b intraveninio gliukozės tolerancijos testo metabolinės P sistemos įgyvendinimų kokybės palyginimas.

Metrika	Apib. srautinis	Apib. kombinacinis	Santykis
$Q_{\mathrm{A}}^{\mathrm{MP}}$	$95,74$	$95,74$	1
$Q_{\mathrm{T}}^{\mathrm{MP}}$	$3,25$	$6,50$	$0,5$
$Q_{\mathrm{R}}^{\mathrm{MP}}$	78.44	$78,42$	1
$Q_{\mathrm{P}}^{\mathrm{MP}}$	$93,16$	91.51	$1,02$
$Q_{\mathrm{I}}^{\mathrm{MP}}$	$33,00$	$33,00$	1
$\overline{Q}_{\mathrm{MP}}$	$37,59$	$43,02$	$0,87$
σ_{MP}	$43,22$	$39,54$	$1,09$

Bendrai apibendrintas kombinacinis įgyvendinimas pasižymi geresne kokybe Q_{MP}. Kaip matosi iš kokybių santykio S4.1 lentelėje, kokybės persvarą labiausiai lemia didesnis skaičiavimo greitis. Apibendrinto kombinacinio įgyvendinimo kokybės dispersija taip pat yra mažesnė. Tai reiškia, kad bendros kokybės komponentės yra išdėstytos tolygiau.

Kokybės pokytis, kai įgyvendinimui naudojamas didesnis žodžio ilgis, pavaizduotas S4.1 pav. Apibendrintas kombinacinis metodas yra naudojamas, nes jis pasižymi aukštesne bendra kokybe. Palyginus pavaizduotas kokybes, galima pastebėti, kad žodžio ilgio pasirinkimas daro pakankamai didelę įtaką įgyvendinimo kokybei.

Norint vienu metu įgyvendinti kuo daugiau IVGTT MP sistemų, galima naudoti didelės talpos LPLM, tokią kaip Virtex-7 XC7VX690T. Pritaikius šį LPLM lustą, vienu metu ta pati LPLM gali skaičiuoti iki 42 apibendrintų MP sistemų.

Atsižvelgiant į praktinį IVGTT sistemų panaudojimą gliukozės kiekio kraujyje stebėjimui, iš viso vienoje LPLM galimų įgyvendinti sistemų skaičius apibrėžiamas taip:

$$N_{\mathrm{IVGTT}} \triangleq \frac{f_{\mathrm{val}}}{N_{\mathrm{val}}} \times N_{\mathrm{MP}} \times t_{\mathrm{mon}}\,, \tag{S4.1}$$

čia N_{IVGTT} – bendras skaičiuojamų IVGTT sistemų skaičius, f_{val} – duomenų generavimo greitis, N_{val} – bendras reikšmių skaičius diskretaus laiko ašyje, N_{MP} – vienoje LPLM galimas įgyvendinti MP sistemų skaičius, t_{mon} – gliukozės ir insulino stebėjimo sensoriaus atskaitų intervalas.

Apibendrinto kombinacinio 32 b įgyvendinimo f_{val} reikšmė lygi 13 MHz, o IVGTT sistemų N_{val} yra 200. Virtex-7 XC7VX690T LPLM vienu metu galima įgyvendinti 42 apibendrintas IVGTT sistemas. Gliukozės stebėjimo sensoriai paprastai generuoja suvidurkintas reikšmes kas 3–5 minutes, o gliukozės sensoriaus signalo pasikeitimo intervalas yra sekundžių eilės. Todėl toliau naudojamas 1 s intervalo reikšmė. Pritaikius šiuos duomenis (S4.1) formulėje, bendras skaičiuojamų IVGTT sistemų skaičius yra:

$$N_{\mathrm{IVGTT}} = \frac{13 \cdot 10^6}{200} \times 42 \times 1 = 2.73 \cdot 10^6\,. \tag{S4.2}$$

Šie rezultatai rodo, kad įmanoma paskaičiuoti virš 2 milijonų IVGTT sistemų vienu metu, panaudojant vieną galingą LPLM. Žinoma, nėra atsižvelgta į kitas funkcijas, kurias

turėtų atlikti LPLM realiame įgyvendinime, tačiau tai parodo, kad MP sistemų įgyvendinimas panaudoja nedidelę dalį LPLM išteklių ir gali būti pritaikomas nebrangiuose mažos galios elektroniniuose įrenginiuose.

Bendrosios išvados

Metabolinės P sistemos matematinio modelio įgyvendinimo aparatinėje įrangoje problema buvo išspręsta. Gauti šie elektros ir elektronikos inžinerijos mokslo sričiai reikšmingi rezultatai:

1. Taikant pasiūlytą metabolinių P sistemų įgyvendinimo metodiką visas šešias žinomas ir disertacijoje nagrinėtas intraveninio gliukozės tolerancijos testo metabolines P sistemas galima apibendrinti viename LPLM įgyvendinime su ne didesne nei 15% vidutine kvadratine paklaida naudojant ne trumpesnį nei 32 bitų žodžio ilgį.
2. Naujas metabolinių P sistemų jungtinės kokybės matas ir jo grafinis atvaizdas aiškiai atskiria įvairių metabolinių P sistemų lauku programuojamų loginių matricų įgyvendinimo kokybę ir sudaro sąlygas atskirai analizuoti šioms sistemoms būdingas kokybės charakteristikas.
3. Pasiūlytas apibendrintas kombinacinis intraveninio gliukozės tolerancijos testo metabolinių P sistemų įgyvendinimo būdas užtikrina nuo 2 iki 3 kartų didesnę spartą lyginant su apibendrintu srautinio įgyvendinimo būdu.

Ateityje šiame darbe pristatytus tyrimus reikėtų tęsti išplečiant modeliuojamų IVGTT sistemų rinkinį pasitelkiant realius eksperimentinius medicininius duomenis ir išbandant sukurtą elektroninę sistemą realiuose taikymuose. Sukurti metodai taip pat gali būti pritaikomi naujų tipų metabolinių P sistemų modeliavimui ir kitokių problemų sprendimui.

Subject Index

Annexes[1]

Annex A. Declaration of Academic Integrity
Annex B. The Co-authors' Agreement to Present Publications Material in the Dissertation
Annex C. The Copies of Scientific Publications by the Author on the Topic of the Dissertation

[1]The annexes are supplied in the enclosed compact disc

Darius KULAKOVSKIS

RESEARCH OF METABOLIC P SYSTEM FIELD
PROGRAMMABLE GATE ARRAY IMPLEMENTATION

Doctoral Dissertation

Technological Sciences,
Electrical and Electronic Engineering (T 001)

Darius KULAKOVSKIS

METABOLINĖS P SISTEMOS ĮGYVENDINIMO LAUKU
PROGRAMUOJAMOMIS LOGINĖMIS MATRICOMIS TYRIMAS

Daktaro disertacija

Technologijos mokslai,
elektros ir elektronikos inžinerija (T 001)

2019 04 30. 11,5 sp. l. Tiražas 20 egz.
Vilniaus Gedimino technikos universiteto
leidykla „Technika“,
Saulėtekio al. 11, 10223 Vilnius,
http://leidykla.vgtu.lt
Spausdino BĮ UAB „Baltijos kopija“,
Kareivių g. 13B, 09109 Vilnius